UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO

PAULO VITOR SILVESTRIN

Sistema para gerenciamento de Dispositivos Móveis baseado em Android

Trabalho de Graduação.

Prof. Dr. Alexandre Carissimi
Orientador

Porto Alegre, Novembro de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Silvestrin, Paulo Vitor

Sistema para gerenciamento de Dispositivos Móveis baseado em Android / Paulo Vitor Silvestrin. – Porto Alegre: Graduação em Ciência da Computação da UFRGS, 2013.

67 f.: il.

Trabalho de Conclusão (bacharelado) – Universidade Federal do Rio Grande do Sul. Ciência da Computação, Porto Alegre, BR–RS, 2013. Orientador: Alexandre Carissimi.

1. MDM. 2. Mobile Device Management. 3. Android. 4. App Engine. 5. Google Cloud Message. 6. Gerenciamento. 7. Dispositivos Móveis. I. Carissimi, Alexandre. II. Título.

AGRADECIMENTOS

Agradeço ao professores e funcionários do Instituto de Informática da UFRGS por proporcionarem este curso de Ciência da Computação de alta qualidade. Agradeço principalmente ao professores Alexandre Carissimi, por todo acompanhamento durante este trabalho de conclusão, e ao Valter Roesler, pela oportunidade de trabalhar no seu grupo de pesquisa por mais de 2 anos.

Agradeço a minha família, principalmente meu pai e minha mãe, por terem disponibilizado toda a estrutura necessária para que eu pudesse me dedicar inteiramente à graduação.

Aos colegas de faculdade e aos amigos pelos os momentos de diversão deixando esta jornada muito mais fácil.

A minha querida namorada pela compreensão nos finais de semana e noites onde os estudos e trabalho tiveram que ser prioridade.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
APK	Android application package file
BYOD	Bring Your Own Device
CDMA	Code division multiple access
CPU	Central process unit
ESN	Electronic serial number
GCM	Google Cloud Message
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IMEI	International Mobile Station Equipment Identity
JSON	JavaScript Object Notation
MAM	Mobile Application Management
MDM	Mobile Device Management
MEID	Mobile equipment identifier
MEM	Mobile Email Management
NoSQL	Not Only Structured Query Language
RAM	Random Access Memory
REST	Representational State Transfer
ROM	Read-only Memory
SaaS	Software as a Service
SDK	Software Development Kit
TI	Tecnologia da Informação
TTL	Time To Live

URL	Uniform Resource Locator
XML	Extensible Markup Language

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O uso de dispositivos móveis dentro de empresas em ambientes de trabalho vem aumentando a cada ano. Com o aumento do poder computacional desses dispositivos, empresas estão migrando aplicações que antes eram apenas utilizadas em computadores de mesa para aplicações móveis. Então, a cada dia novos smartphones e tablets chegam às empresas onde os departamentos de TI tem a necessidade de gerenciar e monitorar o uso destes aparelhos. Este trabalho consiste na especificação e implementação de um sistema para gerenciamento de dispositivos Android. O sistema tem o objetivo de oferecer as funcionalidades básicas de controle do inventário, monitoramento, instalação remota de aplicativos e *chat* com os dispositivos através de uma interface web unificada.

Palavras-chave: MDM, Mobile Device Management, Android, App Engine, Google Cloud Message, Gerenciamento, Dispositivos Móveis.

A Mobile Device Management System based on Android

RESUMO

The use of mobile devices inside companies or work environments has growing with the past of years. With the increase of mobile devices processing power, companies are migrating applications that were used before only in desktops to mobile applications. Then, every day companies buy new smartphones and tablets, the IT department has the task of manage and monitor the usage of these devices. Therefore, this paper shows the specification and implementation of an Android device management system. This system has the aim of provide the basic features of inventory asset control, monitoring, install apps remotely and chat with devices through an unified web interface.

Palavras-chave: MDM, Mobile, Device, Management, Android, Google, Cloud, Message, App, Engine.

1 INTRODUÇÃO

Nos últimos anos cada vez mais empresas estão adotando smartphones e tablets como dispositivos essenciais para muitas tarefas. Hoje a maior parte do uso desses dispositivos é para acessar a Internet ou ler email, mas muitas empresas já estão utilizando para executar programas mais complexos como gestão de negócios, clientes e finanças. Então, esses dispositivos, sendo eles da empresa ou pertencentes ao próprio funcionário, estão muitas vezes conectados na rede interna e acessando dados importantes. Portanto, eles passaram a ser um problema para os departamentos de tecnologia da informação (TI) que devem ter o controle e dar segurança aos dados e equipamentos usados na empresa.

Fazer o controle, segurança e dar suporte técnico aos dispositivos móveis é um desafio que os departamentos de TI das empresas estão tendo. Os dispositivos móveis, diferente dos computadores convencionais, possuem uma grande variedade de plataformas, não tem localização fixa, o departamento de TI pode nunca ter acesso ao dispositivo físico e, além disso, podem facilmente ser perdidos, ou roubados, contendo dados da empresa.

Para auxiliar departamentos de TI no monitoramento e controle desses dispositivos móveis foram desenvolvidas ferramentas conhecidas como MDM (*Mobile Device Management*). MDMs são ferramentas que oferecem controle do inventário dos dispositivos, monitoramento, configuração remota, entre outras funcionalidades. Por exemplo, para a segurança dos dados disponibilizam serviços de criptografia para transmissão e armazenamento de arquivos e emails da empresa, remoção de todos os dados da empresa de forma remota para caso de furto do dispositivo. Para a distribuição e monitoramento de aplicativos da empresa essas ferramentas oferecem serviços semelhantes a *Apple Store* e *Google Play*, onde os aplicativos internos da empresa podem ser instalados, removidos e configurados remotamente de forma segura e transparente.

No mercado existem vários sistemas que oferecem recursos necessários para o gerenciamento dos dispositivos móveis e muitos outros serviços para controle e gerenciamento dos dispositivos e aplicativos. A grande desvantagem dessas soluções é o seu custo elevado.

1.1 Objetivos do Trabalho

As empresas de pequeno e médio porte possuem uma quantidade menor de dispositivos e, normalmente, nem tem a necessidade de todas as funcionalidades que os MDMs tradicionais oferecem. Além disso, o custo dessas ferramentas é elevado para esse perfil de empresa.

O objetivo deste trabalho é implementar uma solução MDM orientada a soluções de código aberto visando um sistema de baixo custo e de fácil configuração.

Para atingir o objetivo proposto, inicialmente, será feito um estudo das funcionalida-

des básicas de um sistema MDM em especial aquelas voltadas a empresas de pequeno porte. Com base nesse estudo, propor e implementar uma arquitetura MDM que atenda as necessidades da equipe de TI dessas empresas para o monitoramento e controle de sues dispositivos móveis.

1.2 Organização do Trabalho

Este trabalho está dividido em 6 capítulos, incluindo esta introdução. O capitulo 2 apresenta um estudo sobre MDM, descrevendo as funcionalidades que devem estar presente nesse tipo de sistema, os desafios encontrados para o desenvolvimento de um MDM e uma visão geral de algumas ferramentas disponíveis no mercado. O capítulo 3 apresenta a especificação de um sistema MDM chamado *Devices Manager* com funcionalidades básicas que atendem as necessidades de uma empresa de pequeno porte. Na sequencia, O capítulo 4 descreve como o sistema MDM proposto foi implementado, incluindo técnicas, *frameworks* e a plataforma móvel utilizada. No capítulo 5 é feita a avaliação do sistema implementado quanto as funcionalidades e consumo de bateria e dados. Por fim, no capítulo 6, é feita a conclusão do trabalho e idéias para trabalhos futuros.

2 MOBILE DEVICE MANAGEMENT - MDM

O gerenciamento de dispositivos móveis (Mobile Device Management - MDM) é uma área administrativa que vem crescendo nos últimos anos com o aumento do uso de smartphones e tablets dentro de empresas. Esses dispositivos móveis oferecem acesso a e-mail, documentos compartilhados e aplicações da empresa de qualquer lugar aumentando a agilidade na realização de tarefas. Portanto, é importante o controle desses dispositivos para que não aconteça perda, roubo ou mal uso causando prejuízos à empresa.

Antigamente, celulares e PDAs, por serem limitados eram muitas vezes desprezados pelos departamentos de TI. Hoje em dia, os dispositivos móveis possuem um grande poder computacional e as empresas estão vendo cada vez mais seus funcionários acessando dados importantes através deles. Em vez de insistirem que seus funcionários não utilizem seus próprios dispositivos para trabalho muitas empresas estão adotando o modelo BYOD "*bring your own device*". Esse modelo habilita funcionários a usarem seu próprio dispositivo tanto para fins de trabalho quanto pessoais, mas esse modelo exige gerenciamento eficiente para garantir segurança dos dados da empresa. Portanto, as equipes de TI estão sentindo a necessidade de utilizar sistemas para gerenciamento de dispositivos móveis para oferecer segurança e controle a esse patrimônio da empresa.

Os desafios encontrados no gerenciamento do dispositivos móveis são diferentes dos normalmente encontrados pelas equipes de TI. Como é mostrado na tabela 2.1 (MIKE OLIVER, 2008), os dispositivos móveis possuem algumas limitações comparados com aparelhos *desktop*. Essas limitações tem que ser levadas em conta na hora de planejar o uso e gerenciamento desses aparelhos.

Este capítulo faz uma introdução ao ecossistema MDM descrevendo sua arquitetura, o padrão *open mobile alliance*, funcionalidades básicas e a análise de três sistemas comerciais com uma breve comparação entre eles.

Tabela 2.1: Comparação entre Desktops e Dispositivos Móveis

	Desktops	**Dispositivos Móveis**
Internet	Ilimitada	Limitada
Conectividade	Garantida e confiável	Instável
Suporte Técnico	Suporte Local aos Usuário	Sem suporte local
Acesso ao Dispositivo pelo TI	fácil e assegurado	não assegurado
Plataforma	Usam a mesma plataforma	Variedade de plataformas
Segurança	Segurança de um estabelecimento	Perdido ou Roubado facilmente

2.1 Introdução

Os MDMs são softwares ou serviços que lidam com a segurança, monitoramento, integração e gerenciamento de dispositivos móveis que inclui smartphones, tablets e laptops, de um ambiente de trabalho. O objetivo do MDM é otimizar as funcionalidades e segurança dentro da empresa e ao mesmo tempo proteger a rede corporativa. Controlando e protegendo os dados e configurações de todos os dispositivos móveis, os MDMs podem reduzir os custos de suporte e reduzir riscos dentro de uma empresa.

Outra área de gerenciamento que está associada ao MDM é MAM (*Mobile Application Management*). Os MAMs são sistemas responsáveis pela autorização e controle de acesso à aplicativos desenvolvidos internamente, ou disponíveis comercialmente, utilizados por empresas em seus dispositivos ou em dispositivos dos funcionários (BYOD). Os sistemas de *Mobile Application Management* possibilitam o departamento de TI instala os aplicativos necessários, controlar o acesso aos dados da empresa, remover dados importantes armazenados no dispositivo em caso de perda ou do funcionário não trabalhar mais na empresa (IT, 2010).

Existem ainda outras especializações relacionadas a MDM como: *Mobile Content Management* - MCM para distribuição de documentos da empresa dentro de um ambiente seguro; *Mobile e-mail Management* - MEM para controlar que dispositivos podem acessar o e-mail da empresa e prevenir perda e roubo de e-mails e anexos com dados importantes.

2.2 Arquitetura básica de um MDM

A arquitetura tipica de um MDM inclui um módulo servidor, um cliente e um centro para configuração remota. O servidor é quem envia comandos de gerenciamento ao dispositivo. O cliente é executado no dispositivo, ele recebe e implementa os comandos enviados pelo servidor. O centro para gerenciamento remoto é um console administrativo utilizado para atualizar ou configurar qualquer dispositivo. Um dos principais recursos de softwares MDM são recursos *Over-the-air* (OVERAIR, 2013), que é a habilidade de configurar remotamente um dispositivo móvel ou um grupo de dispositivos móveis; enviar aplicativos e atualizações; bloquear ou limpar os dados do dispositivos.

Existem soluções MDM nos modelos *Software as a service - SaaS* e *on-premise*. Como as tecnologias móveis estão em constante evolução, sistemas SaaS (baseados em nuvem) são mais fáceis, baratos e rápidos para a realização de atualizações, enquanto soluções *on-premise* requer que a empresa tenha hardware próprio e preocupação com atualizações e manutenção.

2.3 Open Mobile Alliance (OMA)

A *Open Mobile Alliance (OMA)* (OMA, 2013) é uma organização que visa regulamentar padrões abertos para a indústria de dispositivos móveis. A OMA normatiza protocolos aplicados em qualquer tecnologia de rede celular utilizada para promover comunicação e transferência de dados. A aderência as normas é completamente voluntária, a OMA não tem papel mandatário. Entre os vários comitês técnicos da OMA está a OMA DM - (OMA *Device Management*) que especifica funcionalidades de gerenciamento para dispositivos móveis. A tabela 2.2 mostra a lista dos diferentes grupos de trabalho e comitês que pertencem a OMA junto com uma breve descrição.

Tabela 2.2: Grupos de Trabalho e Comitês da OMA

Comitê	**Responsabilidade**
Architecture	Define a arquitetura geral da OMA, fiscaliza e auxilia os outros grupos de trabalho.
Communications (COM)	Define os conjuntos de funcionalidades e métodos que serão utilizados como meio de comunicação entre diferentes aplicações móveis.
Content Delivery	Escreve e mantêm especificações relacionadas a tecnologia *Push*.
Device Management	Define protocolos de gerenciamento e mecanismos para proporcionar um gerenciamento robusto do ciclo de vida de um dispositivo.
Interoperability	Identifica, especifica e mantem os processos, políticas e programas de teste para garantir a interoperabilidade das especificações da OMA.
Location	Desenvolve especificações para garantir a interoperabilidade de serviços de localização móvel.
Release and Planning Management	Planeja e gerencia os *releases* da OMA.
Requirements	Identifica e especifica os casos de uso, interoperabilidade e requisitos de usabilidade de serviços de todos os grupos de trabalho da OMA.

O OMA-DM, na especificação *Device Management Requirements* versão 1.2 (ALLIANCE, 2007), define um MDM como um *framework* que oferece as seguintes funcionalidades:

- **Provisionamento**: Registro e configuração do dispositivo de maneira automática no primeiro uso.
- **Configuração do Dispositivo**: Permite a alteração de configurações e parâmetros do dispositivo.
- **Atualização de software**: Oferece um meio para atualização e instalação de aplicativos ou softwares do sistema.
- **Gerenciamento de Falhas**: Reporta erros e oferece dados do status do dispositivo.

2.4 Funcionalidades de um MDM

Citar todas as funcionalidades que um sistema de MDM pode ter é uma tarefa quase impossível dada a gama de possibilidades existentes (MATHIAS, 2013). Nesta seção são descritas as funcionalidades básicas que esses sistemas devem ter, segundo (PHIFER, 2013), para permitir que o departamento de TI tenha a capacidade de gerenciar os dispositivos.

Note que as necessidades de cada empresa no gerenciamento de seus dispositivos são diferentes. Portanto, nem todas as funcionalidades listadas aqui são essenciais para todas empresas e nem todas são implementadas por todos os sistemas de MDM. Logo, o departamento de TI de cada empresa deve analisar as funcionalidades que precisam e então escolher o MDM que se adapte as suas necessidades.

2.4.1 Inventário de Dispositivos

Claramente, um MDM deve manter a lista dos aparelhos a serem gerenciados. Essa lista de aparelhos deve incluir alguns dados físicos como identificação do dispositivos,

modelo do hardware, versão do firmware, etc. O inventário deve permitir algum tipo de classificação, ou forma de agrupar os dispositivos. Por exemplo, um MDM pode auto-classificar os dispositivos por versão de sistema operacional, ou por modelo. O MDM deve oferecer uma maneira de atualização do inventário com adição e remoção de aparelhos. Hoje em dia muitos smartphones oferecem serviços de localização, então o inventário também pode ter a localização física dos dispositivos.

2.4.2 Provisonamento de Dispositivos

O gerenciamento de dispositivos inicia com a inclusão ou autorização do dispositivo dentro do MDM. Para isso é importante saber que plataformas o MDM suporta incluindo sistema operacional, fabricante, modelo e versão. Por exemplo, Apple iOS, Google Android (e suas diferentes versões), BlackBerry OS, Microsoft Windows Phone.

Existem diferentes maneiras de instalar o agente MDM no dispositivos. Alguns dispositivos já vem com sistema MDM nativo instalado (Apple iOS, BlackBerry OS). Outros o usuário do dispositivo deve visitar alguma *app store* ou portal do sistema MDM para fazer *download* do agente MDM. Outra possibilidade é enviar por e-mail ou mensagem de texto instruções de *download* e instalação do sistema MDM no dispostivo.

2.4.3 Gerenciamento de Aplicativos

Muitos sistemas de MDM vão além da configuração e inventário de dispositivos. Oferecem ferramentas para distribuição e atualização de aplicativos utilizados pela empresa. Existem também os sistemas MAM que lidam especificamente com o gerenciamento de aplicativos dentro de empresas.

2.4.4 Monitoramento e Suporte

Um sistema MDM tem um grande papel no diagnóstico de problemas nos dispositivos mostrando status em tempo real de memória, bateria, processador, conexão com Internet. Além deste monitoramento, oferece também a possibilidade de controle e configuração remota do dispositivo facilitando o suporte técnico aos usuários. O monitoramento do dispositivo também pode ser salvo na forma de histórico, com isso, MDMs podem ajudar na auditoria e fiscalização do uso dos dispositivos.

2.4.5 Segurança e Proteção de Dados

Segurança é um premissa muito importante em um MDM. Dispositivos móveis podem ser perdidos, ou roubados, facilmente, então o MDM deve ter recursos para proteger os dados da empresa nesses casos. Para isso, oferecem políticas de autenticação para acesso a dados e aplicativos contendo informações sensíveis da empresa e meios para remover remotamente todos os dados importantes em caso de roubo do dispositivo. Outra forma de segurança é bloquear aplicativos, ou recursos do aparelho, considerados inseguros pela empresa.

Para a proteção dos dados, alguns MDM oferecem *backup/restore* automáticos, além de *tracking* de dados, ou seja, manter o histórico de transferência e modificação de arquivo, em caso de auditorias.

2.4.6 Suporte às diferentes plataformas

É importante também que o sistema de MDM tenha suporte para as diversas plataformas móveis que estão disponíveis hoje no mercado como: iOS, Android, BlackBerry,

Windows Mobile, etc. Para empresas que adotam a cultura BYOD é importante que o MDM suporte as mais variadas plataformas móveis.

2.5 Sistemas MDM Existentes

Nesta seção serão descritos alguns dos sistemas de MDM existentes no mercado. O objetivo é fornecer uma visão das funcionalidades disponíveis e o funcionamento dos sitemas MDM mais utilizados do mercado. De acordo com o Instituto Gartner, um grupo de cinco empresas controla aproximadamente 60% do mercado de MDM mundial (GARTNER, 2013). Essas empresas são Good Technology, SAP, AirWatch, MobileIron e Fiberlink. Todos os sistemas encontrados são proprietário e pagos, não foi encontrado nenhum sistema livre ou gratuito.

Desses cinco sistemas de MDM foram escolhidos o AirWatch, Fiberlink e MobileIron que são descritos nas próximas seções.

2.5.1 AirWatch

AirWatch (AIRWATCH, 2013) é a empresa líder no mercado em MDM segundo a Gartner (GARTNER, 2013), possui mais de 1000 funcionários. O AirWatch Oferece todo o tipo de soluções para gerenciamento envolvendo mobilidade para as plataformas Android, Apple iOS, BlackBerry, Mac OS, Symbian, Windows 8/RT/32, Windows Mobile e Windows Phone. A solução de MDM inclui também MAM e MEM(Mobile e-mail Management) e os valores partem de quatro dólares por dispositivo.

A solução da AirWatch tem como argumento resolver os desafios associados a mobilidade oferecendo um meio simples e eficiente de visualizar e gerenciar todos os dispositivos a partir de um administrador central. O sistema possui um console web baseado em HTML5, que pode ser acessado a qualquer momento e lugar, onde, pode ser visualizado todos os dispositivos registrados sendo eles da empresa, do funcionário ou compartilhado, não importando a plataforma ou tipo do dispositivo. A figura 2.1 é uma imagem do console de gerenciamento. O cadastro de dispositivos é simples para todas as plataformas. O usuário deve apenas navegar para uma URL fornecida pelo administrador e, se o agente AirWatch não estiver instalado no dispositivo, ele será redirecionado para a respectiva loja onde pode ser realizado o *download*. Assim que o usuário é autenticado as restrições apropriadas, aplicativos e conteúdo são automaticamente instalados no o dispositivo.

AirWatch também permite a criação de perfis de configuração que podem ser aplicados aos dispositivos automaticamente. Os perfis podem ser associados a dispositivos conforme sua propriedade, sistema operacional, um grupo dentro da organização, ou um usuário individual.

A ferramenta oferece gerenciamento de aplicativos, ou seja, o administrador pode ver todas as aplicações instaladas, tanto as públicas, quanto os aplicativos de uso interno. O AirWatch também permite a distribuição e controle de instalação de aplicativos internos. O administrador pode criar listas de aplicativos proibidos e lista de aplicativos obrigatórios. A ferramenta ainda oferece um SDK para empresas que desenvolvem aplicativos próprios utilizando as funcionalidades de segurança do AirWatch. A figura 2.2 mostra a tela de gerenciamento de aplicativos. A ferramenta de MDM AirWatch é sem dúvida muito completa e de fácil usabilidade. É sempre a ferramenta citada em qualquer busca na Internet por MDM.

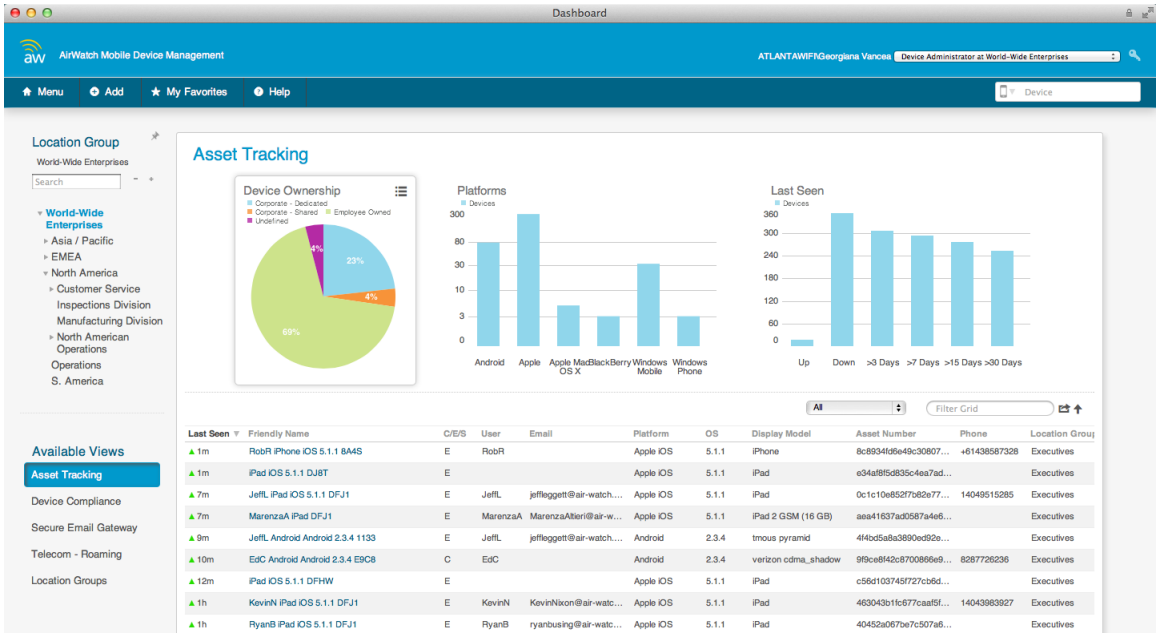

Figura 2.1: Console de Gerenciamento de Dispositivos do AirWatch

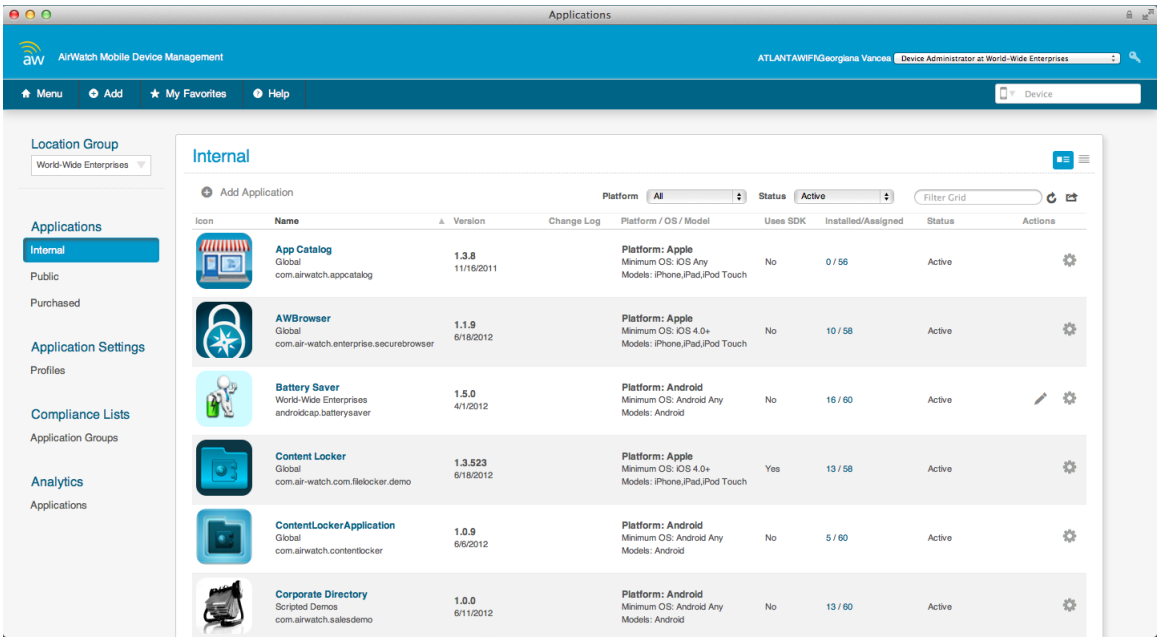

Figura 2.2: Tela de gerenciamento de aplicativos do AirWatch.

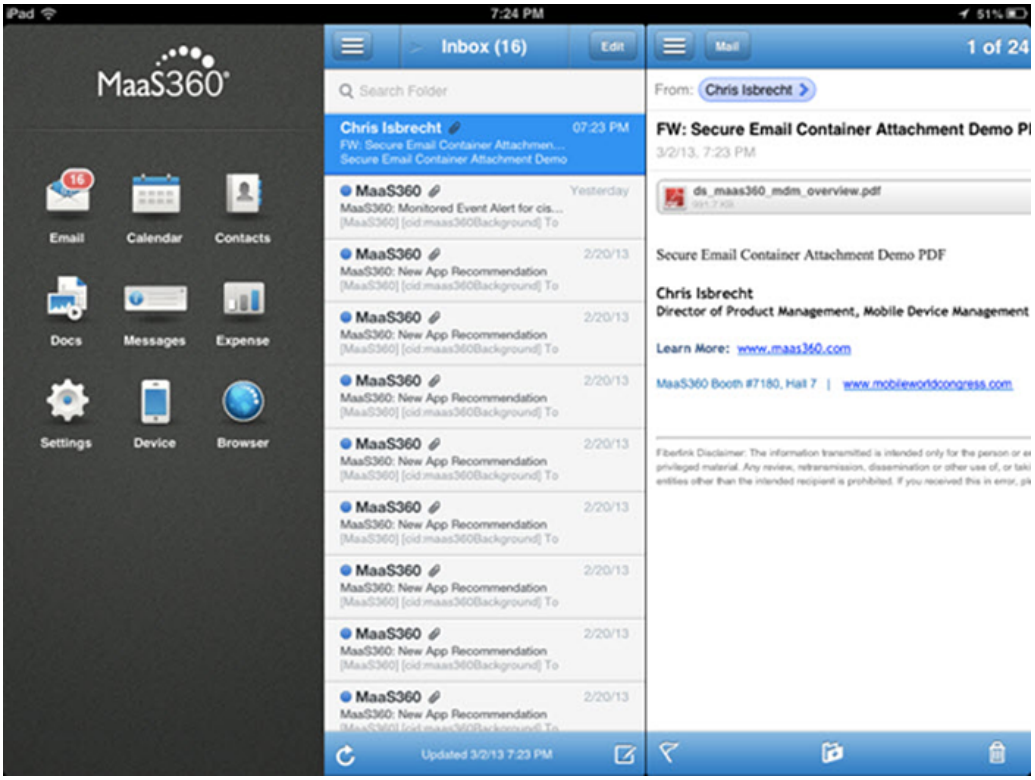

Figura 2.3: Tela inicial do aplicativo MaaS360 para iOS.

2.5.2 Finberlink Maas360

O Maas360 (FIBERLINK, 2013) tem como principal propaganda ser uma plataforma na nuvem, de rápido e fácil desenvolvimento, que oferece uma visão e controle dos dispositivos, aplicativos e documentos da empresa. A sua documentação diz que suporta todos os tipos e plataformas de dispositivos incluindo Kindle Fire e Symbian.

Para cadastro de dispositivo o procedimento é parecido com o AirWatch, porém, além do compartilhamento da URL, ele oferece a possibilidade de enviar um SMS com as instruções de provisionamento.

O Maas360 também oferece o suporte para uma *App Store* da empresa que permite o gerenciamento dos aplicativos. Ainda, disponibiliza recursos para gerenciamento de conteúdo(MCM). Porém este sistema não oferece uma solução *on-premise*, o que pode ser um problema para empresas que não estão confortável com o modelo de coputação em nuvem.

A figura 2.4 apresenta o controle unificado a partir do qual o administrador tem acesso a todos os detalhes do dispositivo, incluindo sua configuração. As figuras 2.3 e 2.5 mostram o aplicativo cliente do Maas360 para iOS e Android, respectivamente.

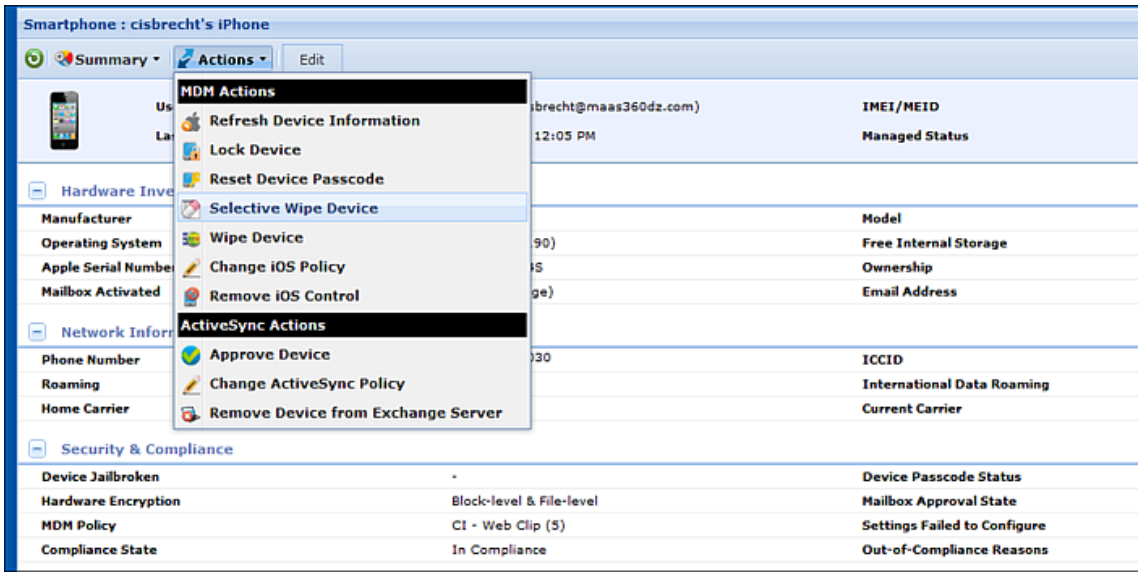

Figura 2.4: Tela de detalhes do dispositivo do MaaS360.

Figura 2.5: Tela inicial do aplicativo MaaS360 para Android.

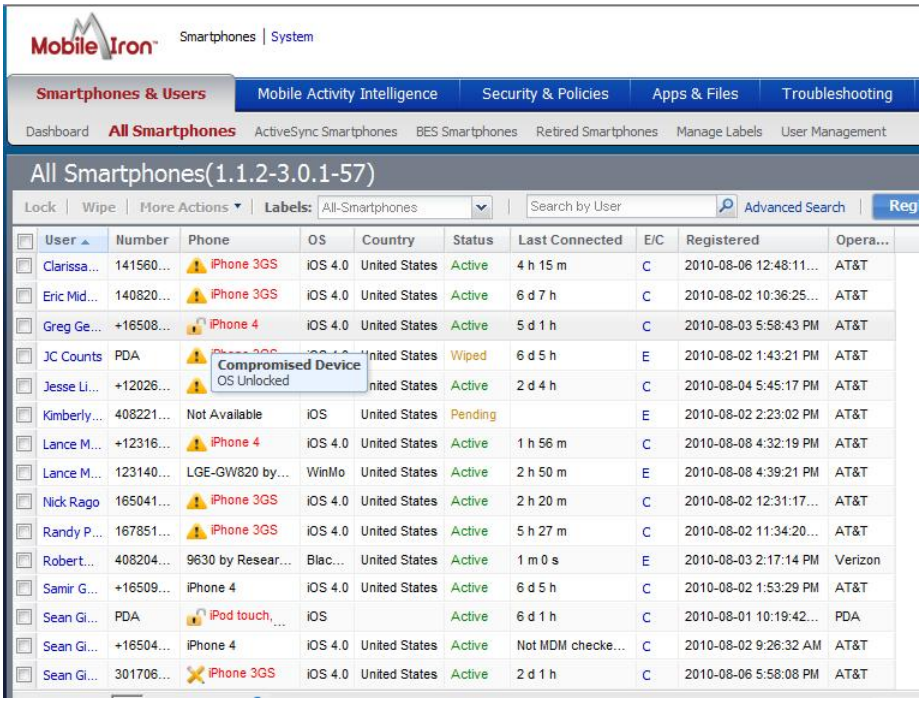

Figura 2.6: Tela do console de gerenciamento do MobileIron para visualização dos Dispositivos.

2.5.3 MobileIron

O MobileIron (MOBILEIRON, 2013a) é uma plataforma que oferece recursos para gerenciamento e segurança de dispositivos, aplicativos e documentos para empresas de todos os tamanhos. Esta solução oferece como diferencial uma configuração fácil e rápida tanto no modelo *on-premise* quanto no modelo *software as a service - SaaS*. A solução *on-premise* é uma aplicação fácil de instalar que pode ser inserida na rede corporativa e pronta para usar em poucas horas de trabalho(MOBILEIRON, 2013b).

Essa ferramenta oferece o gerenciamento de dispositivos como parte da plataforma de gerenciamento disponibilizando serviço de e-mail, compartilhamento de conteúdo, gerenciamento de certificados de segurança e de identidade, e suporta todos os sistemas operacionais modernos. O MobileIron também tem facilidades para empresas que adotam o modelo BYOD, como a disponibilidade de um portal, onde o próprio funcionário pode acessar para fazer o provisionamento do seu aparelho.

A figura 2.6 mostra a tela do console de gerenciamento onde é possível ter uma visão geral de todos os dispositivos cadastrados da empresa, permitindo também buscar por um usuário, bloquear ou limpar o aparelho, e outras funções. A figura 2.7 mostra gráficos e dados sobre os dispositivos cadastrados e sobre atividades de provisionamento. A figura 2.8 é um *screenshot* de uma tela do aplicativo iOS mostrando a interface cliente do gerente de aplicativos, nessa tela o usuário pode instalar aplicativos das empresa ou aplicativos da *App Store*.

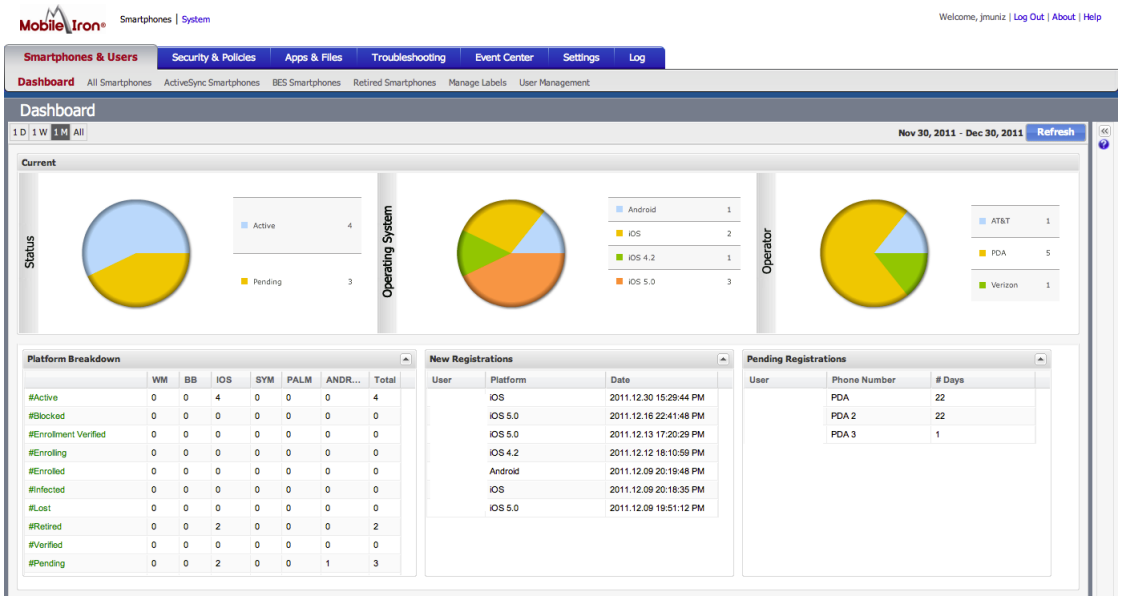

Figura 2.7: *Dashboard* do MobileIron para visualização dos dispositivos.

Figura 2.8: Tela do aplicativo iOS MobileIron de gerenciamento de aplicativos.

2.6 Comparação

Os sistemas MDMs do mercado estão se tornando verdadeiros canivetes suíços com o suporte para inúmeras funcionalidades. A tabela 2.3 fornece uma comparação entre os sistemas *AirWatch*, *Maas360* e *MobileIron* quanto as funcionalidades oferecidas, plataformas suportadas e algumas características gerais. Esta tabela foi elaborada baseada no trabalho observado em (WORLD, 2013).

Tabela 2.3: Comparação entre Sistemas MDM

Sistema / **Arquitetura**	**AirWatch**	**MobileIron**	**Maas360**
On-Premisse	Sim	Sim	Não
SaaS	Sim	Sim	Sim

Sistema / **Plataformas Suportadas**	**AirWatch**	**MobileIron**	**Maas360**
iOS	Sim	Sim	Sim
Android	Sim	Sim	Sim
Extensões de Android	Knox, Safe	Knox, Safe	Knox, Safe, 3LM, HTC, LG
BlackBerry	Sim	Sim	Sim
Android	Sim	Sim	Sim
Wondows Phone, 7, 8	Sim	Sim	Sim
Symbian	Sim	Sim	Sim
Outros	Mac OS X 10.7 e superiores	Mac OS X	Amazon Kindle Fire, Mac OS X, Linux

Sistema / **Funcionalidades MDM**	**AirWatch**	**MobileIron**	**Maas360**
Proteção de Senhas	Sim	Sim	Sim
Restaurar Senha	Sim	Sim	Sim
Limpeza Remota	Sim	Sim	Sim
Limpeza Seletiva	Sim	Sim	Sim
Bloqueio Remoto	Sim	Sim	Sim
Symbian	Sim	Sim	Sim
Setar VPN, Wi-Fi, APN, proxy-gateway settings	Sim	Sim	Sim
Desabilitar WIFI	Sim	Sim	Sim
Desabilitar Rede de Dados	Sim	Sim	Sim
Provisionamento e Registro Automático	Sim	Sim	Sim
Desabilitar Câmera	Sim	Sim	Sim
Desabilitar Bluetooth	Sim	Sim	Sim
Gerenciar Dispositivos Anexos (impressoras e scanners)	Sim	Não	Não

2.7 Considerações Finais

Como foi visto neste capítulo, sistemas MDM são utilizados pelo departamento de TI de empresas para controle e gerenciamento dos dispositivos móveis. Esses sistemas muitas vezes são de alto custo porém, esse custo é amortizado pela redução nos gastos da empresa com manutenção e suporte aos dispositivos e aplicativos.

Os sistemas encontrados no mercado são bastante parecidos oferecendo uma interface web para o departamento de TI visualizar monitorar e configurar remotamente os dispositivos, além de muitos recursos para o gerenciamento de aplicativos. As diferenças entre eles está na forma de disponibilizar as informações e controles aos usuário, ou seja, as interfaces. Todos tendo bastante ênfase na preocupação com a segurança dos dados da empresa.

Na pesquisa feita por sistemas MDM não foi encontrado nenhuma solução para pequenas e médias empresas, que não possuem muitos aparelhos, e portanto não necessitam de sistemas tão completos. Também não foi encontrado nenhum sistema MDM livre, ou de código aberto, que pode ser um requisito para empresas que querem saber exatamente o que o sistema está executando, como ele funciona e que tenha baixo custo. Esses fatores motivaram o desenvolvimento deste trabalho.

3 DEVICES MANAGER - ESPECIFICAÇÃO

Devices Manager é um sistema que tem o objetivo de auxiliar os administradores de TI ou equipes responsáveis pela tarefa de controle e manutenção e gerenciamento dos dispositivos móveis de uma empresa. O administrador deve saber onde está o aparelho, quem e como está sendo utilizado, manter os aplicativos da empresa instalados e atualizados. Essa é uma tarefa bastante desgastante visto que, a cada dia, as empresas adotam mais tablets e smartphones, e os usuários dos dispositivos muitas vezes não são capacitados para deixá-los pronto para serem usados.

Portanto, o sistema proposto tem como objetivo principal facilitar o gerenciamento desses dispositivos. Esse objetivo é alcançado através de uma interface que permite a configuração e monitoramento dos aparelhos de uma forma totalmente remota.

Este capítulo descreve a arquitetura do sistema proposto neste trabalho e especifica o funcionamento de cada módulo.

3.1 Arquitetura do Sistema

O sistema de gerenciamento de dispositivo móveis, como é mostrado na figura 3.1, é composto por três módulos: servidor, aplicação móvel e aplicação web.

O servidor do sistema é quem faz a comunicação com os dispositivos e a aplicação web. Ele recebe dados atualizados vindos dos dispositivos e envia mensagens a eles através da tecnologia *Push* (PUSH, 2013). A aplicação web faz uso periódico do servidor, pois requisita dados em intervalos determinados de tempo para que o usuário sempre tenha informações atualizadas sem precisar recarregar a página web. O servidor acessa uma base de dados onde ficam armazenadas as informações atuais e históricos dos dispositivos.

A aplicação móvel é quem coleta as informações de uso do aparelho e as envia para o servidor. O objetivo dessa aplicação é, sem comprometer a autonomia da bateria do dispositivo, fazer com que o servidor sempre tenha o status mais atualizado possível do dispositivo. Além disso, oferece um serviço de *chat* para comunicação com os responsáveis pela gerência do aparelho e disponibiliza uma lista com os aplicativos da empresa dando a possibilidade do usuário de instalá-los quando necessário.

A aplicação web é por onde o responsável pelo gerenciamento do dispositivos tem acesso a informações de como e onde o aparelho está sendo utilizado. Essa interface web pode ser acessada de qualquer computador com acesso a Internet. Assim, o administrador pode monitorar os dispositivo mesmo longe de seu local de trabalho.

O sistema possui dois tipos diferentes de usuários: administrador e usuário final. O administrador é quem tem acesso a interface web e todas as suas funcionalidades de configuração dos dispositivos móveis sob gerência. O administrador é aquele que tem conhecimento sobre os aplicativos da empresa; quem deve utilizar os aparelhos e como eles

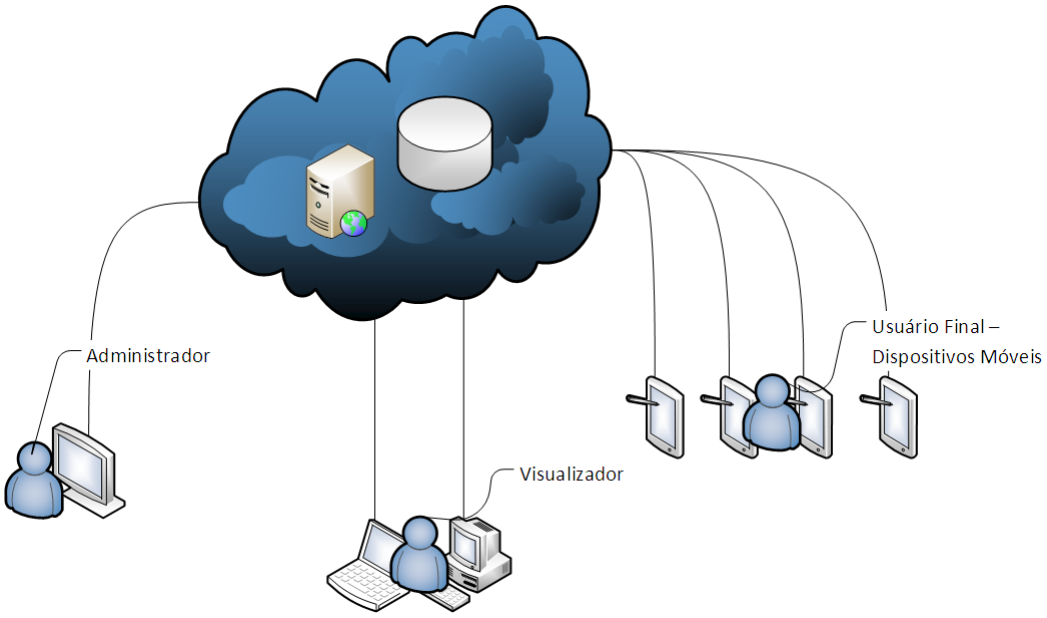

Figura 3.1: Arquitetura do sistema *Devices Manager*.

devem ser utilizados. Além disso, o administrador deve ser capaz de responder dúvidas dos usuários dos dispositivos, muitas vezes resolvendo problemas remotamente através do próprio sistema de gerenciamento.

O usuário final é aquele que emprega o dispositivo móvel como equipamento computacional para realizar suas tarefas dos dia a dia. O usuário final utiliza o sistema de gerenciamento através de um aplicativo específico instalado no dispositivo móvel. O usuário final pode se comunicar com o administrador do sistema através do *chat*. Além disso, o usuário final pode instalar ou atualizar aplicativos da empresa.

O *Device Manager* também auxilia o usuário final que está procurando um *smartphone* ou *tablet* para uso, permitindo que ele acesse o sistema como um visualizador. Como visualizador o usuário final não precisa e não deve ter acesso aos recursos do administrador, ele apenas deve ter acesso ao status atual dos aparelhos para escolher algum para realizar a sua tarefa.

O sistema permite o cadastro de empresa, dispositivo e administradores. O cadastro de empresa é feito na primeira vez que um administrador entra no sistema. Ele deve informar o nome da empresa e criar um **código de registro**. Feito isso a empresa está cadastrada e este administrador está automaticamente cadastrado para esta empresa.

Os dispositivos de uma determinada empresa são cadastrados no sistema utilizando o **código de registro** mencionado anteriormente. O sistema também permite que uma empresa tenha vários administradores que são cadastrados através do seu e-mail.

3.2 Servidor

O servidor é por onde todas as mensagens do sistema passam. Ele tem a missão de receber os dados e as requisições e tratá-los adequadamente. A figura 3.2 mostra como acontece o fluxo de mensagens entre o servidor e seus clientes. As próximas seções explicam como se dá a comunicação entre o servidor, a aplicação web e os dispositivos móveis.

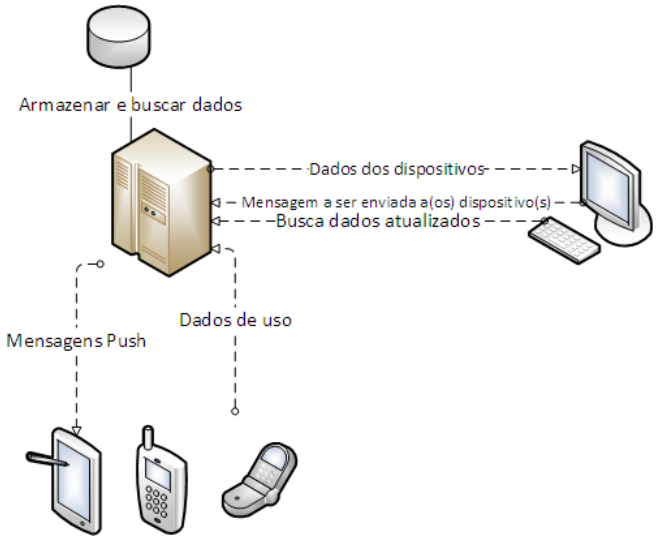

Figura 3.2: Diagrama do fluxo de troca de mensagens do Sistema.

3.2.1 Comunicação entre servidor e Aplicação Móvel

A comunicação entre os dispositivos móveis e o servidor se dá através do aplicativo específico instalado no aparelho, que tem a função de tando enviar quanto receber dados do servidor. O aplicativo envia dados de uso atualizados ao servidor sempre que necessário. Nesse caso o servidor deve tratar os dados e armazená-los no banco de dados.

O dispositivo móvel recebe as mensagens sempre que o administrador, através da aplicação web, requisitar. As possíveis mensagens são as listadas na seção 3.2.2. Essa comunicação poderia ser implementada através de *Polling* como acontece na aplicação web. Porém, como os dispositivos móveis normalmente possuem uma limitação de Internet e bateria, isso não é viável. Portanto, essa comunicação se dá pela tecnologia *Push*, onde o servidor envia a mensagem ao cliente.

3.2.2 Comunicação entre servidor e Aplicação Web

Como será visto na seção 3.4, o cliente web faz um uso frequente do servidor. Além de buscar a página web renderizada, ele faz acessos periódicos (conhecido como *Polling*(POLLING, 2013)) para estar sempre exibindo os dados mais recentes enviados pelos dispositivos móvel sem o usuário ter que atualizar a página. Por exemplo, quando um dispositivo móvel liga o GPS, essa informação é enviada ao servidor, e o administrador, estando com a aplicação web aberta, após alguns segundos, terá essa atualização automaticamente. Com isso, o administrador tem uma experiência em tempo real de monitoramento de seus dispositivos. Em contrapartida, o *Polling* tem um grande consumo de processamento do servidor e de Internet.

Para atender do modo eficiente as requisições periódicas que a aplicação web faz, o servidor deve implementar algum sistema de *cache* das buscas no banco de dados. Muitas dassas requisições periódicas vão retornar dados que não foram modificados, e acessar o banco de dados para atendê-las seria muito ineficiente e custoso. A partir da aplicação web também são mandadas mensagens para o servidor que por sua vez deve enviá-las aos dispositivos. Essas mensagens podem ser: mensagem de *Chat*; requisição para instalar ou desinstalar algum aplicativo; pedido por dados atualizados; habilitar ou desabilitar GPS, *bluetooth*, *wireless*.

3.2.3 Mensagens *Push*

Mensagens *Push* é o nome dado ao tipo de comunicação via Internet onde a requisição para uma dada transação é iniciada pelo servidor e não pelo cliente como normalmente acontece. O objetivo desta tecnologia é fazer com que o cliente receba dados que são destinados a ele sem ele ter que consultar o servidor. Mensagens, ou notificações *push*, são muito utilizadas em aplicações de e-mail, notícias, resultados de esportes, monitoramento de rede de sensores, entre outros.

Para a implementação de mensagens *push* em aplicações para dispositivos móveis os fabricantes do sistema operacional, normalmente, oferecem *frameworks*. Por exemplo: Google Cloud Messaging para aplicações Android, Apple Push Notification Service para aplicações iOS, Windows Azure para aplicações Windows.

3.3 Aplicação Móvel

Essa é a parte do sistema instalada nos dispositivos móveis como smartphones e tablets. A aplicação móvel tem por objetivo ser intuitiva e de fácil utilização, onde a maioria das suas funcionalidades são transparentes ao usuário final, ou seja, devem acontecer sem que ele perceba.

A aplicação móvel pode ser divida em duas partes: uma oferece serviços que permite a interação com usuário final e outra que permite interação com o servidor através de uma série de serviços em *background*.

3.3.1 Serviços para Usuário Final

A parte do aplicativo móvel que permite interação com usuário oferece os seguintes serviços: registro do dispositivo, *chat* onde o usuário final pode se comunicar diretamente com o administrador, e uma área que permite instalação e atualização dos aplicativos de uso da empresa.

3.3.1.1 Registro de Dispositivo

O registro do dispositivo é quando acontece o cadastro do aparelho para uma determinada empresa. O registro do dispositivo pode ser feito tanto pelo usuário final quanto pelo administrador. O único pré requisito para o registro é ter em mãos o código de registro. Esse código é aquele escolhido pelo administrador quando faz o cadastro da empresa (seção 3.4.1). É um código único que deve ser utilizado para registrar todos os dispositivos móveis daquela empresa. Para registrar o dispositivo deve-se colocar um nome ao aparelho e o código de registro de dispositivos.

A tela de registro só é apresentada ao usuário final na primeira vez que executa o aplicativo, ou quando é cancelado o registro do aparelho. O cancelamento do registro do dispositivo só pode ser feito pelo administrador via interface web. Dessa forma, mesmo desinstalando o aplicativo do aparelho, ele continuará registrado no sistema e ao reinstalar o aplicativo e executá-lo a tela de registro não será apresentada. Isso se deve ao fato de o código de identificação do dispositivo ficar registrado como pertencente a aquela empresa, então ao executar o aplicativo pela primeira vez ele irá verificar se está registrado enviando o código ao servidor.

O código único de identificação de dispositivo pode ser o IMEI(GSMA, 2013) para dispositivos GSM e MEID(TIA, 2013a) ou ESN(TIA, 2013b) para dispositivos CDMA. Para dispositivos sem módulo telefônico utiliza-se um código de identificação fornecido pelo sistema operacional.

3.3.1.2 Chat

O serviço de *chat* é um canal de comunicação direta com a equipe de TI responsável pelos aparelhos. Logo, quando o usuário do dispositivo estiver com algum problema com o aparelho, ou dúvida na utilização de algum aplicativo, ele pode utilizar esse recurso para se comunicar com pessoal de TI da empresa.

Esse canal de comunicação pode ser visto como uma forma de registro de problemas e dificuldades que os funcionários da empresa estão tendo com o uso dos aplicativos e dispositivos. As mensagens ficam salvas no sistema se tornando um histórico de problemas reportados. Então, o histórico pode ser utilizados na hora de planejar o treinamento dos funcionários dando mais ênfase para as dificuldades reportadas por este sistema.

3.3.1.3 Gerenciamento de Aplicativos

O gerenciamento de aplicativos é o serviço que permite realizar o *download* e a instalação dos aplicativos de uso específico da empresa, assim como listá-los. Essa funcionalidade é importante para o caso de algum funcionário estar em campo e perceber que alguma aplicação que ele necessita no momento não está instalada no dispositivo, ou está desatualizada. Isso pode acontecer por falha da equipe de gerência ou por esse dispositivo não ser o designado para essa tarefa. Mas, com essa funcionalidade, se o dispositivo estiver com acesso a Internet, ele pode instalar a aplicação e realizar a sua tarefa.

Nesse exemplo citado, o usuário também poderia comunicar a equipe responsável pelo dispositivo para realizar a instalação remotamente. Porém, é importante permitir que o usuário possa instalar o aplicativo, se isso for necessário e ele for capacitado.

Esse serviço deve possuir uma lista onde cada linha contêm o ícone, nome e tamanho das aplicações disponíveis pela empresa, além de um botão informando se a aplicação está instalada ou precisa ser instalada ou atualizada. Portanto, estando o usuário com Internet disponível, com apenas um toque na tela ele terá a sua disposição o aplicativo que ele precisa.

3.3.1.4 Fluxo das Telas

A figura 3.3 mostra o fluxo das telas que fazem parte do aplicativo móvel. O aplicativo inicia com a tela de registro, se o dispositivo ainda não estiver registrado no sistema de gerenciamento. Caso contrário o sistema inicia na tela de serviços. A tela de serviços nada mais é do que uma forma de navegar para a tela de *chat* (seção 3.3.1.2) ou tela de aplicativos (seção 3.3.1.3).

3.3.2 Serviços de Controle

Os serviços executados em *background* são o centro deste aplicativo. Esses serviços, devem responder as requisições por dados atualizados (sincronização) mesmo com o dispositivos estando em modo *idle*. São esses serviços que permitem o controle, localização e configuração remota do dispositivo.

A sincronização significa montar um pacote com todos os dados de uso do dispositivo, de cada serviço, e enviar ao servidor esse pacote. Dessa maneira, evita-se fazer várias

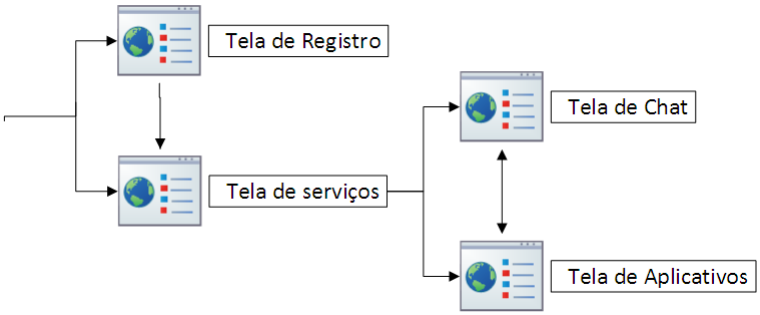

Figura 3.3: Fluxo das telas da aplicação móvel.

conexões com o servidor para enviar pequenas quantidades de dados, minimizando o gasto de bateria e tráfego de rede.

3.3.2.1 Controle

Como controle do dispositivo entende-se o administrador saber como o dispositivo está sendo utilizado. Para isso, ele deve ter acesso a dados como:

- **Status de GPS, rede *wireless*, *bluetooth* e Internet móvel:** Esses status devem ser gravados no momento que são ativados ou desativados. O status deve ser salvo junto com a data que aconteceu a mudança.

- **Utilização de dados de rede:** Nos dispositivos podem ter uma interface de Internet móvel, ou seja, a rede 3G ou 4G, e outras interfaces de rede como *wireless*. Para o administrador do aparelho é importante ter o controle desses dados de uma maneira diferente, pois muitas vezes a quantidade de tráfego permitida pela interface de Internet móvel é limitada. A quantidade de dados enviados e recebidos pela interface de Internet móvel e total (por todas as interfaces de rede) serão registradas e enviadas ao servidor. Os valores específicos de uso de rede de cada aplicativo, também será registrado, porém, ficará armazenada no dispositivo e será enviada ao servidor apenas quando uma solicitação especifica por esses dados for recebida. O serviço responsável pelo monitoramento do uso da rede deve fazer a leitura dos dados.

- **Nível da Bateria:** O nível da bateria deve ser registrado sempre que o mesmo apresentar alguma alteração. O valor armazenado é mandado para o servidor. Não é necessário guardar o histórico dos dados, quando o nível da bateria sofre alguma alteração o valor antigo pode ser descartado mesmo sem ter sido enviado. Esse dado é importante para um usuário poder escolher um aparelho com bateria suficiente para realizar a sua tarefa; ou para o responsável colocar o dispositivo para carregar quando o nível de bateria está baixo.

- **Ociosidade do Dispositivo:** Para detectar se o aparelho está sendo utilizado, ou ocioso, deve-se checar se a tela está ligada ou desligada e se o aparelho está em movimento através do sensor acelerômetro. Esses são alguns dos fatores indicativos de que o aparelho está, ou não, em uso. Esse serviço deve ter uma informação precisa e atualizada quando for solicitada uma sincronização.

- **Status de CPU e memórias RAM, ROM e SD card:** Informar o status atual de cada um desses recursos de hardware no momento da sincronização com o servidor.

Ou seja, qual a porcentagem da CPU está sendo utilizada e a quantidade de memória ocupada.

3.3.2.2 Localização

A capacidade de facilmente detectar e oferecer dados de localização para os aplicativos se tornou uma das maiores funcionalidades das plataformas móveis de hoje. Os dispositivos móveis, em sua maioria, oferecem recursos que disponibilizam a posição do aparelho e um indicativo da precisão dessa posição, também conhecida como qualidade da posição.

A localização dos aparelhos é uma informação importante que o administrador deve ter para o registro dos locais onde os dispositivos estiveram. Hoje, a maioria dos tablets e smarthphones vem equipados com serviços de localização utilizando GPS, antenas de telefonia e rede *wireless*. Para maiores detalhes sobre como funciona os serviços de localização em dispositivos móveis o leitor pode buscar no trabalho de Millet & Stoud (MILETTE; STROUD, 2012).

O aplicativo móvel deve ter um serviço eficiente que possibilite a localização e rastreamento do dispositivo. O hardware de GPS tem um grande consumo de energia, porém esse serviço não poderá comprometer a durabilidade da bateria. As posições coletadas, junto com data e a sua qualidade, devem ser salvas em um banco de dados local e enviadas ao servidor sempre que acontecer uma sincronização.

Em um ambiente onde encontram-se vários dispositivos semelhantes, ou no caso de extravio de aparelho, a função *encontre-me* pode facilitar o administrador a encontrar um aparelho especifico. Portanto, a aplicação também deve suportar essa funcionalidade que é a seguinte: quando chega uma requisição *encontre-me*, imediatamente o dispositivo deve emitir um alerta sonoro para facilitar sua localização.

3.3.2.3 Configuração Remota

A configuração remota dos dispositivos é uma das mais importantes funcionalidades do sistema. Isso possibilita que o administrador dê manutenção ao dispositivo sem necessitar que tenha acesso físico ao mesmo. Essa facilidade permite que, através de uma interface web, o administrador possa:

- **Instalar, atualizar e desinstalar aplicativos:** Para instalar, ou atualizar um aplicativo, o serviço irá receber uma requisição contendo o URL onde é possível fazer o *download* desse aplicativo para então instalá-lo. Isso deve acontecer em *background*, sem o conhecimento do usuário. Para remover uma aplicação o serviço recebe uma requisição com o nome do aplicativo que o administrador deseja remover.

- **Rede *wireless*:** O serviço deve permitir ligar e desligar a interface de rede *wireless*. Além disso, quando solicitado, deve prover a lista de redes *wireless* disponíveis e fazer a conexão na rede escolhida pelo administrador.

- **GPS e rede móvel:** Habilitar e desabilitar esses recursos quando solicitado.

3.4 Aplicação Web

A interface web é o ambiente que permite visualizar, analisar e gerenciar todos os dispositivos registrados. É através dela que o administrador pode monitorar e realizar a configuração remota dos aparelhos. Para descrever melhor a aplicação web pode-se dividi-la nas seguintes funcionalidades: autenticação, registro, inventário, gerência de aplicativos, configuração, *chat* e localização.

3.4.1 Autenticação

Os serviços de autenticação propostos neste sistema são simples. A tela de autenticação apresenta duas formas para o usuário entrar no sistema: para o administrador através de um provedor *OpenId* e para o usuário final através do código de registro.

O administrador autentica-se no sistema através de sua conta *OpenId*. *OpenId* é um padrão aberto de autenticação que permite usuário se autenticarem através de provedores de identidade, alguns exemplos de provedores são Google, Yahoo e Facebook. Portanto, ao entrar na aplicação web como administrador, ele será transferido para a página de autenticação do seu provedor onde ele deve permitir que o sistema *Devices Manager* tenha acesso a alguns dados de sua conta, como nome e e-mail.

O usuário final se autentica na aplicação web através do código de registro obtido com o administrador. Ao se autenticas com o código de registro, o usuário pode somente visualizar o inventário de dispositivos cadastrados no sistema. Isso atende aqueles funcionários que desejam localizar um aparelha para uso, ou, apenas tem uma visão geral das condições dos dispositivos cadastrados no sistema. Nessa opção, o usuário final assume o papel de visualizador.

3.4.2 Registro

No primeiro acesso do administrador ele deve registrar a sua empresa. Para isso, deve informar os dados da empresa e criar o **código de registro** que será utilizado para registar dispositivos que deseja gerenciar. Depois de informado os dados o sistema verifica se o código de registro escolhido está disponível. Em caso afirmativo, o registro da empresa é confirmado, se não o administrador deverá escolher outro código.

O administrador também pode registrar outras pessoas para serem administradores. Para isso, ele deve informar o e-mail do administrador que será convidado.

3.4.3 Inventário

Informar o inventário dos dispositivos cadastrados é uma das funcionalidades básicas de um sistema MDM. No sistema *Devices Manager* informações sobre o inventário são encontradas na lista de dispositivos, lista de dispositivos em espera e informações do dispositivo.

3.4.3.1 Lista de Dispositivos

A lista de dispositivos tem por objetivo ser um *dashboard* para monitoração dos aparelhos. Nela não é permitido alterar configurações do dispositivos, apenas uma forma de visualização rápida do status atual. Os dados de cada dispositivo exibidos na lista são:

- Nome do Aparelho,
- Nome do responsável,

- Modelo do dispositivo,
- Nível da bateria,
- Status da rede *wireless*, Internet móvel e GPS,
- Mostrar se o aparelho está em uso ou ocioso.

Além desses dados, a interface da lista de dispositivos deve apresentar um botão *encontre-me*, que ao ser ativado o dispositivo, se estiver *online*, deve emitir algum sinal sonoro para facilitar a localização.

3.4.3.2 Lista de Dispositivos em Espera

A lista de dispositivos em espera é importante para que o administrador tenha controle dos dispositivos que estão sendo registrados. Um aparelho entra para a lista de espera no momento em que ele é inserido a primeira vez no sistema. A lista exibe o nome do usuário, modelo do aparelho e versão do sistema operacional e uma para opção para aceitar e outra para rejeitar esse dispositivo. Com isso o administrador pode aceitar um dispositivo, assim ele passa para a lista de registrados, ou, o administrador pode rejeitar, nesse caso o dispositivo sai da lista de espera e uma notificação é enviada ao aparelho.

3.4.3.3 Informações do Dispositivo

As informações do dispositivo disponibilizadas são: modelo e fabricante do aparelho; modelo, fabricante e velocidade da CPU; quantidade de memória RAM e memória interna; resolução da tela; versão do Sistema operacional; e versão do Núcleo.

Além disso, o nome e e-mail de quem o dispositivo está registrado também são informados. O administrador também pode atribuir um nome a esse dispositivo ou ao usuário do dispositivo.

3.4.4 Gerência de Aplicativos

Para o gerenciamento de aplicativos o sistema oferece um cadastro e a possibilidade de instalar e remover aplicativos remotamente de um determinado dispositivo. , O cadastro de aplicativos da empresa possibilita adicionar e remover os arquivos de instalação dos aplicativos. Por exemplo, aquivos APK no caso de aplicativos Android. Apenas o administrador tem permissão para isso. Esta funcionalidade é importante, pois apenas os aplicativos que foram previamente adicionados podem ser instalados remotamente pelo administrador nos dispositivos. Logo, todos os aplicativos de uso da empresa devem ser cadastrados e mantidos atualizados.

O sistema também deve disponibilizar a lista de todos os aplicativos instalados em um determinado dispositivos e um botão possibilitando que o administrador remova um determinado aplicativo.

3.4.5 Configuração de Dispositivo

Esse serviço permite o administrador ter acesso aos detalhes de configuração e de uso do aparelho. As informações podem ser estáticas ou dinâmicas. As esticas são aquelas que uma vez definidas não se alteram diante do uso do equipamento como, por exemplo, modelo e versão do sistema operacional. Já as informações dinâmicas são modificadas de acordo com a utilização do equipamento. As principais informações fornecidas por esse serviço são:

- **Status do hardware:** Entre os status de hardware incluem: informação do uso instantâneo do processador (incluindo o número de processos em execução) e memória RAM; o nível da bateria e se ela está ou não carregando; quantidade de espaço utilizado da memória ROM e *SD card*. Também é possível checar o status de ligado ou desligado do *bluetooth*, GPS, *wireless* e rede móvel. O sistema também deve permitir habilitar e desabilitar essas funcionalidades remotamente. Essas informações são bastante dinâmicas, portanto, a interface deve disponibilizar um botão para solicitar informações mais recentes. Para receber a solicitação e responder o dispositivo precisa estar conectado à Internet.

- **Uso de dados:** A quantidade de dados enviados e recebidos pelas interfaces de rede do aparelho é um dos principais indicativos da forma que o dispositivo está sendo utilizado. Portanto sistema fornecerá a quantidade total de bytes trafegados e a quantidade de bytes trafegados pela interface de rede móvel (pois muitas vezes tem limite de tráfego pago pela companhia).

3.4.6 Chat

O *chat* é o local para a troca de mensagens entre o administrador e o usuário final. As mensagens são enviadas ao aparelho e recebidas através da Internet, ou seja, uma mensagem enviada será recebida no dispositivo tão logo ele ficar *online*.

O *chat* pode ser utilizado pelo usuário para reportar alguma dificuldade no uso do aparelho, ou para o administrador enviar algum alerta ou dicas de uso. As mensagens enviadas e recebidas devem ser armazenadas na forma de histórico.

3.4.7 Localização

A localização dos aparelhos é uma das principais funcionalidades do sistema, pois saber onde o dispositivo se encontra auxilia na segurança do patrimônio da empresa.

Por exemplo, com o auxilio de um mapa e um pino é possível mostras a posição mais atualizada disponível de cada dispositivo. Ao clicar nesse pino, abre um balão com nome do usuário e modelo deste aparelho. Se o administrador clicar no balão ele irá navegar para as configurações do dispositivo. Na área de configurações do dispositivo o administrador pode ver o histórico de localização escolhendo o intervalo de tempo. Então todas as posições registradas dentro desse período são plotadas no mapa.

3.4.8 Fluxo das Telas

A figura 3.4 mostra o fluxo de telas da aplicação web do sistema *Devices Manager*. A aplicação inicia com a tela de autenticação que é por onde o administrador ou usuário final entram no sistema. O administrador, ao entrar no sistema, se não possuir uma empresa cadastrada, é redirecionado para a tela de registro onde ele registra a sua empresa e cria o código de registro para essa empresa.

A tela inicial é onde se tem uma visão geral dos dispositivos. Deve mostrar a lista de dispositivos, lista de dispositivos em espera e um mapa mostrando a localização dos dispositivos. Essa tela também deve permitir a navegação para a tela de configuração do dispositivo, tela de cadastro de aplicativos e tela de administração da conta.

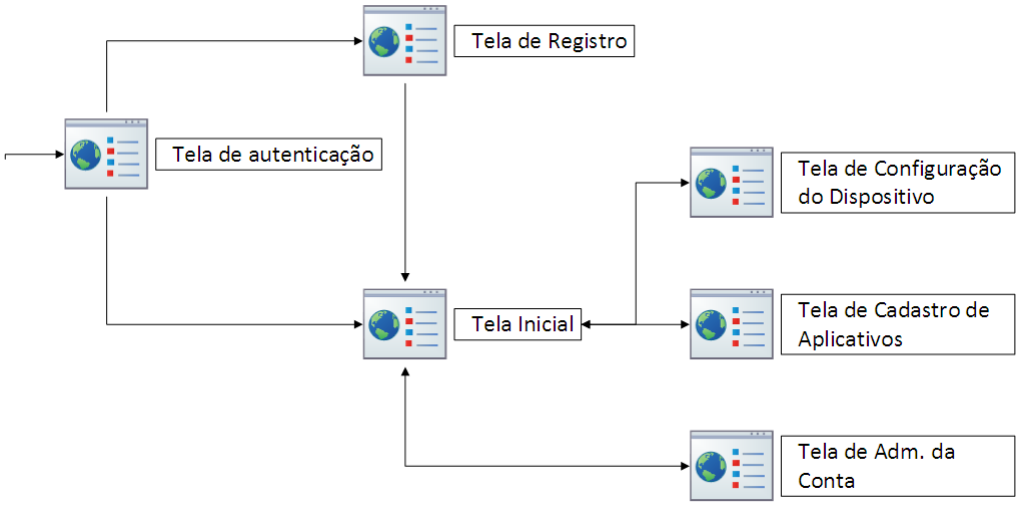

Figura 3.4: Fluxo das telas da aplicação web.

A tela de configuração do dispositivo mostra o detalhes do aparelho, status de hardware, lista de aplicativos instalados, *chat*, ativar e desativar recursos remotamente e instalar e remover aplicativos remotamente. A tela de cadastro de aplicativos é onde são adicionados e removidos os arquivos de instalação dos aplicativos da empresa. Por fim, na tela de administração da conta é possível visualizar os dados da conta e registrar novos administradores.

3.5 Considerações Finais

O sistema *Devices Manager* especificado neste capítulo tem como objetivo ser um sistema MDM (*Mobile Device Management*) que possui funcionalidades que permite monitorar e configurar remotamente dispositivos móveis de uma empresa de pequeno e médio porte. Além disso, ele oferece um serviço de *chat* muitas vezes não previsto em sistemas desse tipo.

Os principais pontos levados em consideração na hora de especificar o sistema foi o consumo de bateria do dispositivo e oferecer ao administrador informações em tempo real de uso do dispositivo. Porém, para o administrador ter essas informações em tempo real o aplicativo precisa coletar dados de uso e enviá-los com frequência e isso pode gerar um alto consumo de bateria, principalmente informações de localização providas pelo GPS. Portanto a implementação desse sistema, que é apresentada no próximo capítulo, tem o desafio de conciliar esses dois parâmetros.

4 IMPLEMENTAÇÃO

Este capitulo apresenta a implementação do sistema *Devices Manager* conforme a especificação fornecida no capítulo anterior. Para sua implementação foi escolhida a plataforma Android, por se tratar de um sistema de código aberto, com boa ferramentas e documentação, além de disponibilizar de maneira fácil os dados necessário para o gerenciamendo do dispositivo móvel.

As próximas seções explicam, em detalhes, como cada módulo foi implementado, as tecnologias utilizadas e o porquê de sua escolha. A seção 4.1 descreve da arquitetura do sistema implementado. A base da comunicação entre os módulos que comões o sistema é o *Google Cloud Messaging (GCM)*, o qual é descrito na seção 4.2. O servidor do sistema é descrito na seção 4.3 onde justifica a escolha do *framework App Engine* do Google. A seção 4.4 explica como foi implementado o aplicativo Android e as técnicas utilizadas para capturar e enviar as informações sem um consumo excessivo de bateria. A seção 4.5 mostra as tecnologias utilizadas na aplicação web para oferecer ao usuário dados dos dispositivos em tempo real. E, por fim, considerações gerais sobre a implementação do sistema.

4.1 Arquitetura geral do Sistema

A figura 4.1 retoma a arquitetura do sistema mostrada na figura 3.1 indicando a tecnologia utilizada em cada módulo. A arquitetura do *Devices Manager* possui um servidor, um cliente e um centro de configuração remota, ou seja, a arquitetura básica de um sistema MDM como descrita no capítulo 2.

Na implementação do sistema foi escolhido utilizar um modelo SaaS onde toda a parte de servidor e banco de dados ficam hospedados nos servidores do Google permitindo às empresas clientes apenas o acesso aos serviços. O cliente é um aplicativo Android que implementa as ações enviadas pelo centro de configuração remota. O centro de configuração remota é uma interface web acessível de qualquer navegador de Internet.

4.2 Google Cloud Messaging (GCM)

Enviar mensagens a partir de um servidor para dispositivos móveis não é uma tarefa fácil. Smartphones e Tablets estão sujeitos a conexões com Internet instáveis, falta de bateria, entre outros problemas que tornam difícil essa comunicação. Uma solução é fazer com que o aparelho, quando estiver *online*, envie mensagens periódicas ao servidor para checar se possui alguma mensagem para ele. No entanto, essa solução é muito onerosa em termos de consumo de bateria e de envio de dados pela Internet para que o dispositivo tenha, em tempo real, as mensagens destinadas a ele.

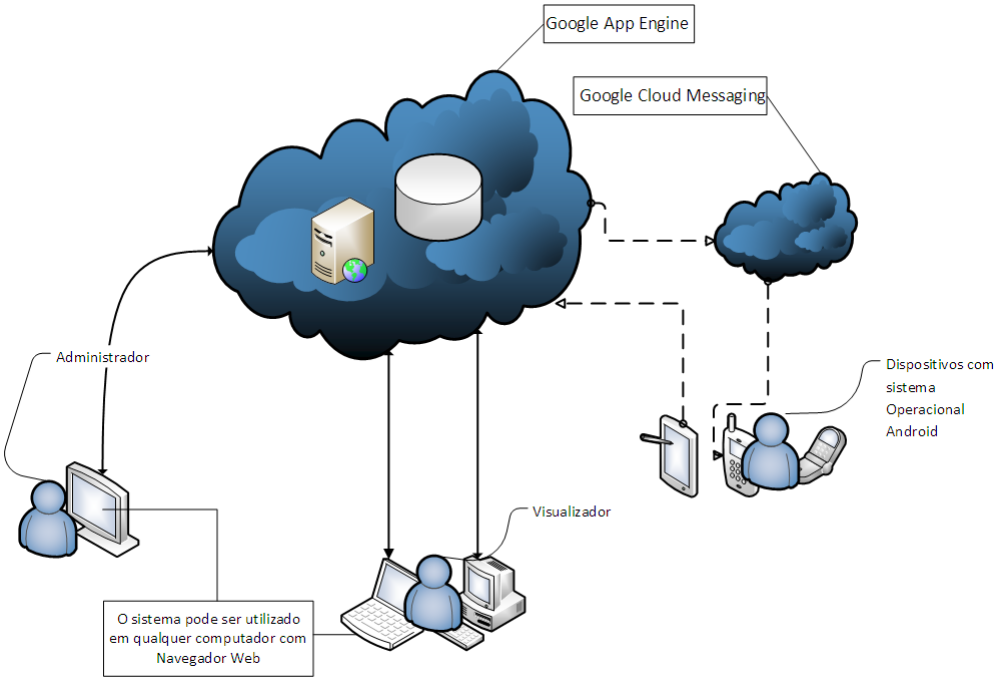

Figura 4.1: Arquitetura da implementação do sistema *Devices Manager*

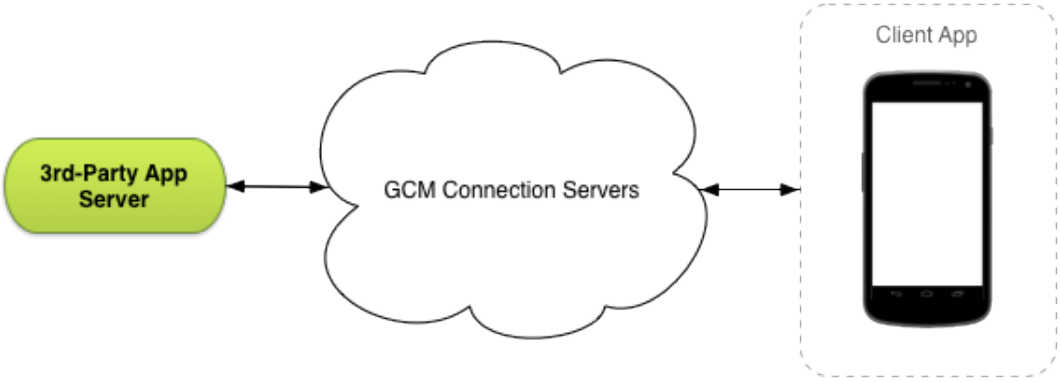

Figura 4.2: Arquitetura do *Google Cloud Messaging* (GCM, 2013).

Para resolver esse problema dos desenvolvedores de aplicativos Android, o Google disponibiliza o *Cloud Messaging* (GCM). Anteriormente, esse serviço era provido pelo *Cloud to Device Messaging Framework (C2DM)* que foi oficialmente descontinuado em Junho de 2012.

4.2.1 Arquitetura do GCM

O GCM é o serviço do Google responsável pelo envio de dados aos dispositivos Android. Sua documentação pode ser encontrada na página para desenvolvedores do Android (GCM, 2013). A figura 4.2, retirada da documentação oficial da ferramenta, mostra a arquitetura do sistema. Esse serviço está disponível para qualquer aparelho com sistema operacional Android 2.2, ou mais recente, que tenha o pacote de serviços do Google instalado.

O *Cloud Message* funciona baseado em uma conexão TCP entre o servidor do GCM e o dispositivo. O servidor GCM mantém um socket TCP na porta 5228 a espera de

solicitações de abertura de conexão (estado de *listen*). Quando o dispositivo está com acesso a Internet ele abre essa conexão TCP com o servidor. Essa conexão é mantida mesmo quando o aparelho está em modo *idle*. O CGM possui um protocolo para envio de mensagens de maneira eficiente e altamente confiável.

Para acontecer essa comunicação são necessárias algumas credenciais de autenticação para que o GCM reconheça o servidor, o aparelho e o aplicativo:

- **ID do Servidor:** é obtido no console da API do Google no momento em que a aplicação é registrada. É utilizado no processo de registro para identificar o servidor que está autorizado a enviar mensagens aos dispositivos.

- **ID da Aplicação:** é a identificação da aplicação móvel que vai receber a mensagem, ou seja, é o nome do pacote da aplicação.

- **ID de Registro:** é a identificação do aparelho junto à aplicação. É fornecido pelos servidores do GCM. Deve ser enviado ao servidor do sistema que por sua vez vai utilizar para enviar mensagens ao dispositivo.

A mensagem enviada ao GCM é composta pelos seguintes parâmetros que indicam como a mensagem será tratada pelo servidor GCM:

- ***registration id*:** esse parâmetro é o identificador do dispositivo que irá receber a mensagem. Pode ser informado uma lista de dispositivos.

- ***delay while idle*:** é um valor booleano que quando *true* indica que a mensagem deve ser enviada ao dispositivo apenas quando ele estiver no estado ativo.

- ***collapse key*:** é um identificador da mensagem. É utilizado no caso em que há mais de uma mensagem a ser enviada para o mesmo dispositivo. As mensagens com o mesmo valor de identificação, na fila de envio, serão colapsadas em uma só.

- ***time to live*:** indica o tempo de vida da mensagem na fila de espera para envio ao destino final (dispositivo). Aceita valores de 0 segundos a 4 semanas.

- ***data*:** é a área onde são inseridos os dados a serem enviados na mensagem. Pode ter no máximo 4KB de tamanho.

O GCM entrega a mensagem ao dispositivo assim que ele ficar *online*. Caso o dispositivo não fique *online* dentro do tempo de vida da mensagem, ela é descartada.

O cliente GCM implementa a tarefa de receber mensagens do servidor GCM e as distribui para os aplicativos instalados no dispositivo. Essa distribuição é feita na forma de *broadcast* de mensagens, no Android chamadas de *Intents* (DEVELOPERS, 2013a), onde o destinatário final(aplicação) verifica se a mensagem é de seu interesse através da categoria do *Intent*. Então, cada aplicação implementa um *Broadcast Receiver* (DEVELOPERS, 2013b) para receber as mensagens e encaminhar para módulo do aplicativo que irá tratá-la.

4.3 Servidor

O servidor é o centro do sistema. Ele deve estar sempre disponível para receber e enviar informações. O servidor normalmente é algo essencial para qualquer sistema, mas no caso do *Devices Manager*, ele acaba ganhando uma importância maior, pois os dispositivos móveis nem sempre estão *online* e com uma boa disponibilidade de Internet. Então, quando estão aptos a mandar dados ao servidor ele não pode falhar.

Por isso, foi feita a escolha pelo *Google App Engine* que é um serviço de *cloud* oferecido pelo Google. Esse serviço oferece uma disponibilidade de 99.95%, segundo sua documentação (GOOGLE, 2013). Além disso, já registrou 0% de indisponibilidade durante um ano(BLOG, 2013). O Google App Engine é um *framework* fácil de usar para desenvolvimento e publicação de serviços web, que disponibiliza uma cota grátis do serviço que é suficiente para os testes deste sistema.

Nesse servidor foi desenvolvido um *web service* que é por onde os dispositivos móveis e clientes web enviam e recebem dados. O *web service*, banco de dados e *framework* de cache utilizados são descritos a seguir.

4.3.1 Google App Engine

O *Google App Engine* é uma ferramenta que permite executar aplicações web na infraestrutura do Google. É uma solução fácil de manter e escalável deixando o programador livre para focar apenas no desenvolvimento da aplicação em si. Suporta aplicações escritas em várias linguagens como JAVA, PYTHON, PHP, GO. Para este trabalho foi escolhido utilizar a linguagem JAVA, pois o autor esta mais familiarizado com essa linguagem.

A aplicação web no Google é executada em um ambiente seguro com acesso limitado ao sistema operacional. Isso isola a aplicação em um ambiente que é independente de hardware, sistema operacional e localização física do servidor. Isso permite distribuir as requisições web entre múltiplos servidores, alocando e desalocando servidores para se adaptar as demandas de tráfego. Essa adaptação dos recursos conforme a sua utilização é o que se denomina de elasticidade na área de computação em nuvem.

Aplicações executando no *App Engine* apenas podem ser acessadas por requisições HTTP ou HTTPS em suas portas padrões. Não é permitido escrever no sistema de arquivos em tempo de execução, ou seja, apenas se tem acesso aos arquivos carregados no *deploy* da aplicação. O código da aplicação só pode ser executado em resposta a alguma requisição, ou tarefa agendada, e precisa responder em até 60 segundos caso contrário o processo é terminado. A aplicação interage com o ambiente por meio de *Java Servlet* (ORACLE, 2013a), e também foi utilizado neste trabalho a tecnologia *JavaServer Pages* (ORACLE, 2013b) para a renderização de páginas web dinâmicas.

O App Engine também oferece um banco de dados NoSQL e *Memcache*(MEMCACHE, 2013) para acesso a dados em alta velocidade. Além disso, suporta integração do aplicativo com o Google Accounts que facilita a autenticação de usuários.

4.3.2 Web Service

Utilizando a ferramenta App Engine foi implementado um *Web Service* para a comunicação com os dispositivos que compõem o sistema. Para implementação foi utilizado o modelo *RESTful* de *Web Service*(RICHARDSON; RUBY, 2007), que é implementado utilizando HTTP e os princípios REST.

Foi escolhido JSON(*JavaScript Object Notation*) como formato de dados que será tro-

cado entre o *web service* e seus clientes. O JSON é derivado da forma como a linguagem JavaScript representa suas estruturas de dados e *arrays*, embora JSON seja independente de linguagem e possui *parser* disponível para muitas linguagens. Esse formato de dados vem sendo bastante utilizado hoje em dia por ser uma alternativa, que apresenta um sobrecusto de dados e processamento menor que o XML. Assim, em aplicações móveis, onde o consumo de dados é algo bastante sensível, JSON é muito utilizado na comunicação com servidores.

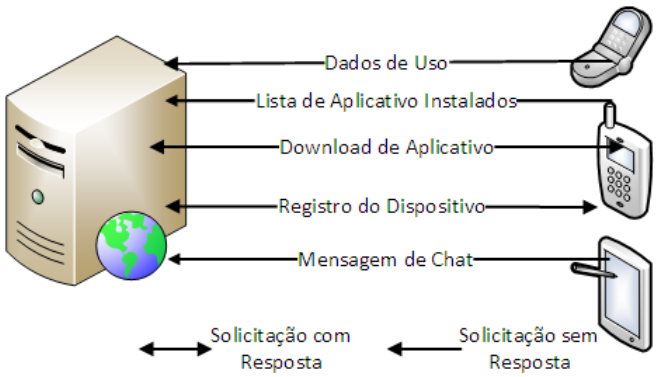

Figura 4.3: Comunicação entre aparelhos Android e servidor através das funções de Web Service.

Foram implementadas funções no *webservice* para atender as necessidades do aplicativo móvel e web. A figura 4.3 mostra as funções de *webservice* que o aplicativo Android faz uso. A figura 4.4 mostra como a aplicação web usa o *webservice* para enviar dados ao servidor. E por fim, a figura 4.5 mostra como a aplicação web usa o *webservice* para buscar dados necessários para montar a página web.

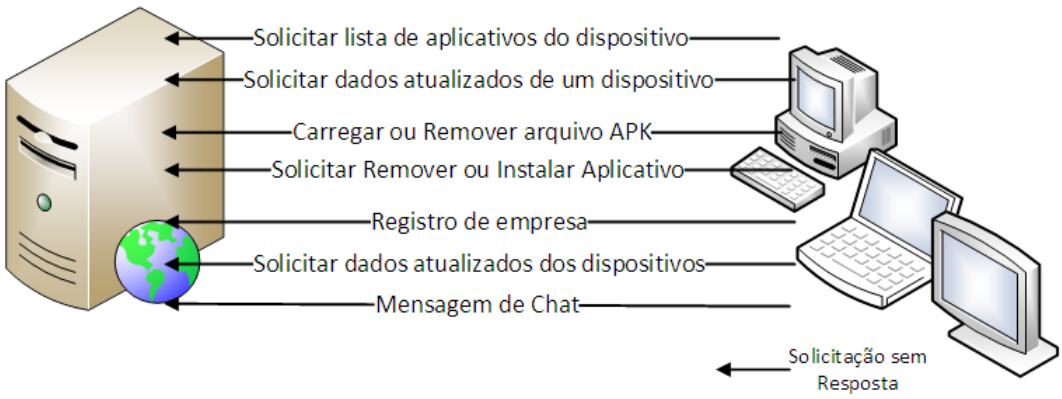

Figura 4.4: Requisições feitas a partir da aplicação web ao servidor.

A seguir são listadas algumas dessas funções dando mais detalhes do funcionamento e implementação.

- Criação e autenticação do administrador: a autenticação acontece através do serviço de autenticação do Google App Engine utilizando a conta do Google do administrador;
- *Upload*, *download* e remoção de arquivos APK (pacote de instalação de aplicativo Android);

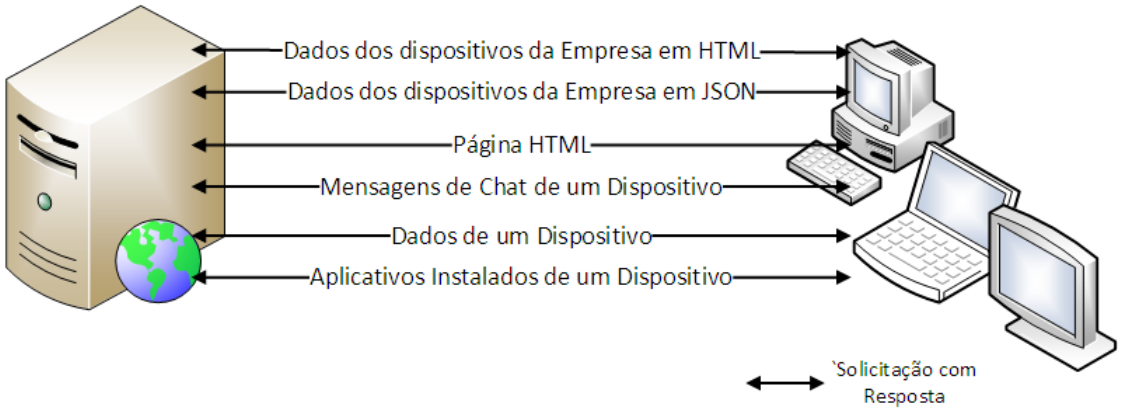

Figura 4.5: Funções do web service por onde a aplicação web busca as informações que serão exibidas.

- Lista dos aplicativos da empresa;
- Requisitar ao aparelho que remova ou instale algum aplicativo;
- Receber mensagens de *chat* vindas da aplicação ou do dispositivo, encaminhar mensagens ao dispositivo e fornecer o histórico de mensagens a partir de uma data;
- Receber a lista de aplicativos instalados em um aparelho e fornecer essa lista quando solicitado;
- Lista de todos os aparelhos de uma empresa com seu status. Essa lista pode ser em formato JSON ou HTML pronto para ser exibido na página web;
- Recebe dados de uso de um dispositivo;
- Registro de aparelho;
- Registra uma empresa com seu nome e código de registro;
- Solicita dados atualizados aos dispositivos de uma empresa;

As seções 4.4 e 4.5 descrevem de que maneira e com que frequência essas funções do *web service* são utilizadas.

4.3.3 Banco de Dados

Como visto anteriormente, o App Engine tem nativamente suporte para banco de dados NoSQL. Diferente de banco de dados relacionais, a natureza do NoSQL, conhecida como *schema-less*, permite que as classes sejam armazenadas assim como elas são. Portanto, muitas das técnicas utilizadas em banco de dados relacionais não se aplicam a um banco de dados NoSQL. Por exemplo, a classe *Company*, que tem como um de seus atributos uma lista de objetos *Device*, pode ser armazenada como uma lista dos IDs desses objetos.

Para facilitar a implementação da persistência dos dados do sistema foi utilizado a API *Objectify*, cuja documentação pode ser encontrada em seu site oficial (OBJECTFY, 2013). Essa ferramenta, além de permitir persistir os modelos de maneira fácil, também já faz uso da *cache*, oferecido pela *App Engine*, de maneira automática.

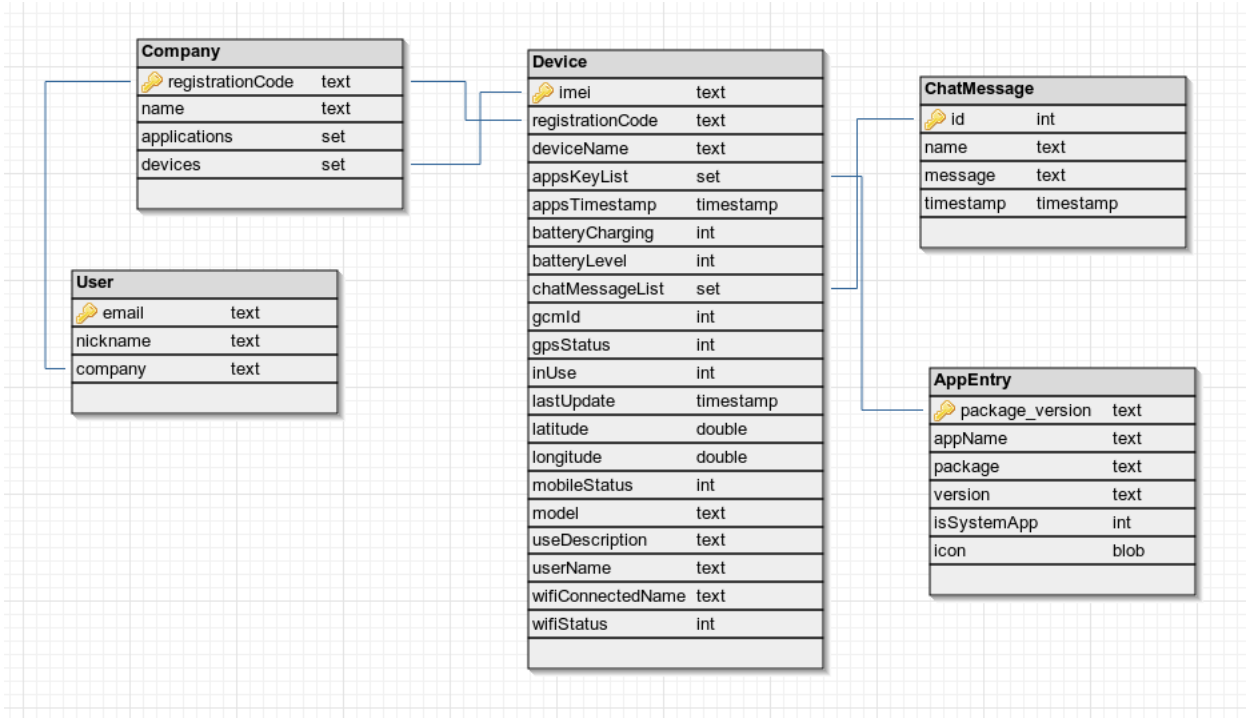

Figura 4.6: Arquitetura do banco de dados NoSQL do sistema.

A figura 4.6 mostra o diagrama do banco de dados resultante do sistema que é composto pelas seguintes tabelas:

- ***Company*:** onde são armazenadas as empresas registradas no sistema. Possui como chave primária o código de registro escolhido pelo administrador. A tabela *Company* também possui o conjunto chamado *Applications* que são os URLs para os aplicativos cadastrados da empresa. Essa tabela também possui o conjunto chamado *Devices* que guarda as referências para todos os aparelhos registrados para a respectiva empresa.

- ***User*:** armazena todos os usuários administradores registrados no sistema.

- ***Device*:** os dados dos aparelhos da empresa são armazenados nessa tabela. Tem como chave primária o IMEI do aparelho. Essa tabela também possui um conjunto com as referências para as mensagens de *chat* (*chatMessageList*) e um conjunto com todos os aplicativos instalados do aparelho (*appsKeyList*).

- ***ChatMessage*:** guarda todas as mensagens de *chat* do sistema. A coluna *timestamp* que é utilizada para ordenar as mensagens por data que foram geradas.

- ***AppEntry*:** nessa tabela são armazenados os aplicativos instalados nos dispositivos. Tem como chave primária o nome do pacote do aplicativo concatenado com a versão. Dessa maneira evita-se que um aplicativo instalado em muitos aparelhos fique armazenado repetidamente na tabela.

4.3.4 Uso do GCM

A figura 4.7 mostra as mensagens onde o GCM é utilizado para notificar o dispositivo. Todas essas mensagens são enviadas com o TTL máximo de 4 semanas, pois é importante

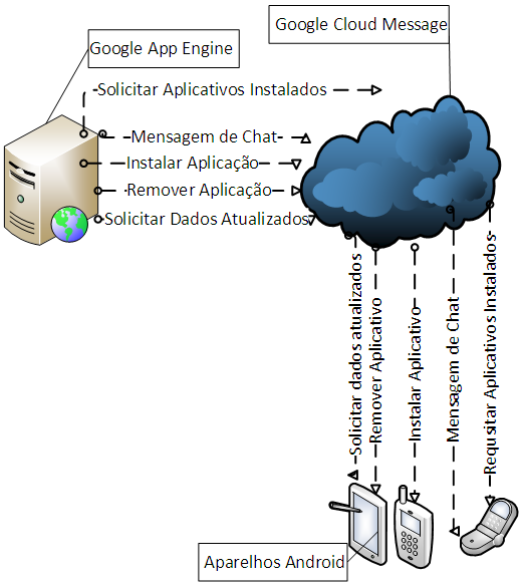

Figura 4.7: Mensagens enviadas aos Dispositivos Android através do *Google Cloud Messaging*.

que todas cheguem ao aparelho. As mensagens que fazem requisição por dados, indicadas na figura como *Solicitar dados atualizados* e *requisitar aplicativos instalados*, são enviadas com o mesmo *collapse key*. Ou seja, se o aparelho ficar *offline* por um longo período, e mais de uma dessas mensagens forem enviadas, apenas a ultima será considerada, as outras serão descartadas. As mensagens *Remover Aplicativo*, *Instalar Aplicativo* e *Mensagem de Chat* tem importância isolada, por exemplo, cada mensagem de *chat* é importante que chegue ao dispositivo. Logo, elas são enviadas com *collapse key* diferentes.

4.4 Aplicação Móvel

Esta seção descreve as funcionalidades e serviços implementados no aplicativo Android, descrevendo as técnicas e ferramentas utilizadas para extrair e enviar os dados de uso do aparelho sem um consumo excessivo de bateria e rede.

O aplicativo Android foi desenvolvido utilizando o SDK nativo do Android, e compilado utilizando a API 17. O aplicativo suporta dispositivos com sistema operacional Android versão 2.3 ou mais recentes.

As próximas seções retomam os serviços para usuário final e serviços de controle, especificados no capítulo 3, adicionando detalhes de como foram implementados.

4.4.1 Serviços para Usuário Final

Os serviços para o usuário final são aqueles que ele tem acesso através de uma interface para interagir com o sistema. O aplicativo do sistema *Devices Manager* implementado possui duas telas principais, no Android chamadas de atividades: a tela para registro do dispositivo, pode ser vista na figura 4.8, e a tela de *chat*, pode ser vista na figura 4.9. A seguir são explicados os detalhes de implementação de cada uma delas.

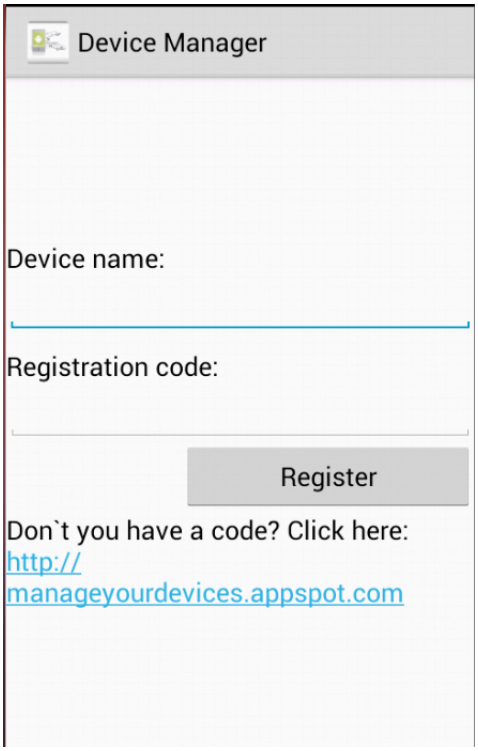

Figura 4.8: Tela para registro do dispositivo.

4.4.1.1 Registro do Dispositivo

A primeira vez que o aplicativo é iniciado em um dispositivo móvel é apresentada a tela de registro. A tela implementada pode ser vista na figura 4.8. Aqui, o usuário fornece um nome ao dispositivo e o código de registro da sua empresa e clica em registrar para dar início ao processo de registro.

Primeiramente, o aparelho se registra no GCM. Para isso ele utiliza o ID do servidor e ID da aplicação, que são valores estáticos no sistema, com esses dados ele se registra no GCM recebendo o código de registro do dispositivo. Feito isso, chama a função do *Web-service* para registro do dispositivo no servidor do *Device Manager* passando os seguintes dados:

- Código de registro do GCM.
- Código de identificação do dispositivo (IMEI). Esse código é obtido através dos procedimentos descritos no artigo (DEVELOPERS, 2013c). O IMEI um identificador único que permite que o mesmo dispositivo nunca seja registrado duas vezes no sistema. Ou seja, mesmo removendo o aplicativo do aparelho ele continuará registrado no *Devices Manager* e quando o aplicativo for reinstalado ele voltará e enviar dados deste mesmo aparelho.
- Código de registro da Empresa que associa o dispositivo móvel à empresa a qual ele pertence.
- Nome dado ao dispositivo móvel.
- Modelo do dispositivo móvel.
- E-mail do usuário registrado no dispositivo móvel.

Feito isso o dispositivo móvel está registrado no sistema e o administrador, pela aplicação web, já pode monitorá-lo.

4.4.1.2 Chat

A funcionalidade de *chat* implementada, exibida na figura 4.9, oferece ao usuário os recursos básicos de enviar e receber mensagens. No caso do aparelho estar *offline* a mensagem fica armazenada e será enviada assim que o aparelho tiver acesso à Internet.

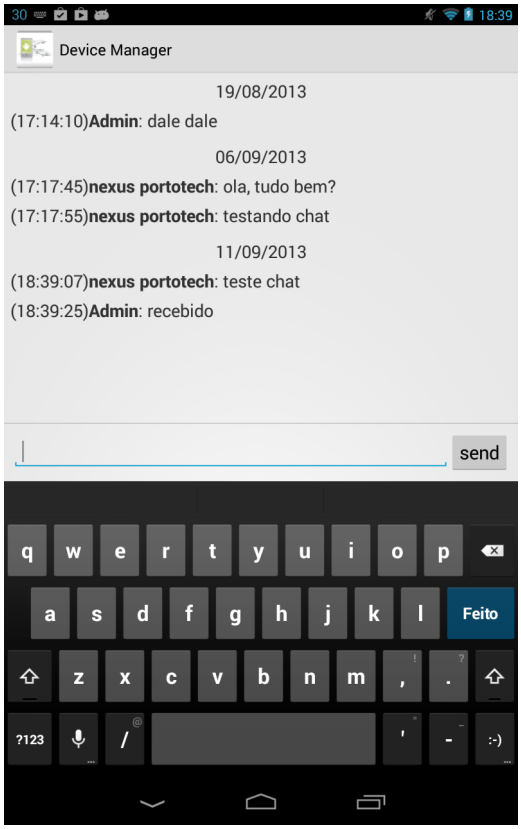

Figura 4.9: Tela de *Chat* do aplicativo *Device Manager*

Para armazenar as mensagens de *chat* recebidas pelo dispositivo móvel foi implementado um banco de dados SQL com a tabela mostrada na figura 4.10. O Android disponibiliza uma API com os recursos necessários para implementação de banco de dados SQLite(DEVELOPERS, 2013d).

O aplicativo implementado possui um serviço chamado *ChatMessagesService* que tem a missão de gerenciar o envio e recebimento de mensagens. Quando uma mensagem

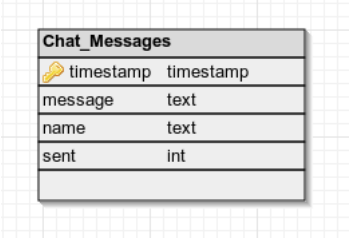

Figura 4.10: Tabela do banco de dados onde é armazenado as mensagens de *chat*.

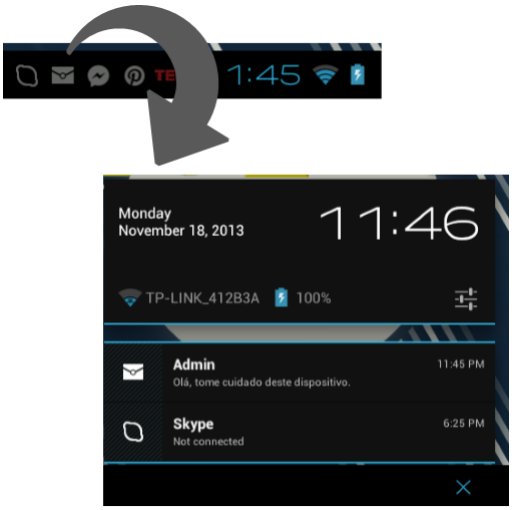

Figura 4.11: Exemplo de como são mostradas as notificações de *chat* no dispositivo Android.

chega via GCM ela é enviada para esse serviço. Então, ele se encarrega de colocar a mensagem no banco de dados e notificar o usuário. Essa notificação pode acontecer de duas maneiras: se o usuário estiver com a tela de *chat* do aplicativo aberta ela será notificada e a lista de mensagens será atualizada; caso contrário uma notificação, como na figura 4.11, é exibida e o usuário pode clicar nela para abrir a tela de *chat*.sec:controle

Inicialmente, para enviar uma mensagem, ela é adicionada ao banco de dados com o campo *sent* marcado como falso. Então é enviado uma notificação ao serviço que deve tentar enviar as mensagens de *chat* pendentes, se o aparelho estiver *online*. Sempre que o dispositivo ficar *online* o serviço também recebe uma notificação para tentar enviar mensagens pendentes.

4.4.2 Serviços de Controle

Essa seção descreve a implementação dos serviços de controle especificados na seção 3.3.2. Esses serviços implementados, que o aplicativo executa em *background*, são os listados na figura 4.12 que apresenta um *screenshot* da tela dos serviços do aplicativo *Device Manager*.

4.4.2.1 Monitoramento

O serviço chamado *HandleFreshDataRequest* é o responsável pelo monitoramento dos recursos mais dinâmicos do dispositivo: GPS, bateria, rede *wireless*, Internet móvel, localização e reconhecimento de atividade. Além disso, é responsável por enviar dados atualizados ao servidor quando requisitado por mensagens vindas através do GCM.

O GPS é monitorado apenas quanto ao status, ligado ou desligado. Então, se o usuário ativar ou desativar o GPS, na mesmo momento, o serviço recebe uma notificação e tenta enviar ao servidor o status atualizado.

O serviço recebe notificação quando o aparelho é colocado para carregar a bateria, ou se a bateria entra em um nível muito baixo. Quando um desses eventos acontece é enviado para o servidor o status atualizado do bateria.

Sempre que a rede *wireless* é ativada, desativada, ou conectada a uma rede o serviço tenta enviar ao servidor o status atual.

Dentro do *Google Play Services* há disponível um serviço que oferece a localização

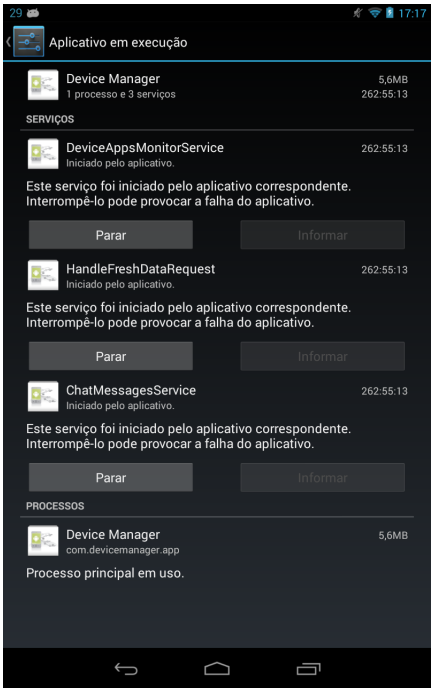

Figura 4.12: Tela que mostra todos os serviços que são executados pelo aplicativo *Device Manager*

e a detecção de atividade do aparelho. O serviço de localização do Google disponibiliza diferentes perfis de consumo de bateria. Neste trabalho, foi configurado para um consumo moderado e uma frequência entre de 1 min e 1 hora. Então a aplicação irá receber as localizações da seguinte maneira:

1. Se o aparelho estiver sendo utilizado apenas dentro da empresa, ou sem muita movimentação, ele irá receber posições através da rede *wireless*, ou 3G, com pouca frequência provavelmente chegando ao intervalo máximo de 1 hora.

2. Se o usuário estiver em campo, o serviço irá detectar nova localização aproximada pelas antenas de telefonia celular, ou redes *wireless* encontradas. Aumentando a frequência de novas posições.

3. Caso o usuário esteja utilizando algum aplicativo de localização, como o Google Maps, com o GPS ativo, ele terá posições de qualidade atualizadas a todo o instante. Nesse caso, a aplicação irá receber posições a cada 1 minuto que é a frequência máxima configurada.

O reconhecimento de atividade do dispositivo é feito com a leitura de sensores de baixo consumo. O serviço tem a capacidade de detectar as seguintes atividades: caminhando, andando de bicicleta, em um veículo, parado, e outras (desconhecidas). Essa leitura acontece dentro de intervalos de 5 minutos. Cada leitura dos sensores retorna uma provável atividade com um valor de probabilidade de acerto. Então, sempre que a atividade detectada tiver probabilidade superior a 50%, e for diferente da anterior, essa informação é enviada ao servidor.

Mais informações sobre o funcionamento do *Google Play Services* pode ser encontrada na documentação oficial sobre localização (LOCATION, 2013) e sobre o reconhecimento de atividade (RECOGNITION, 2013).

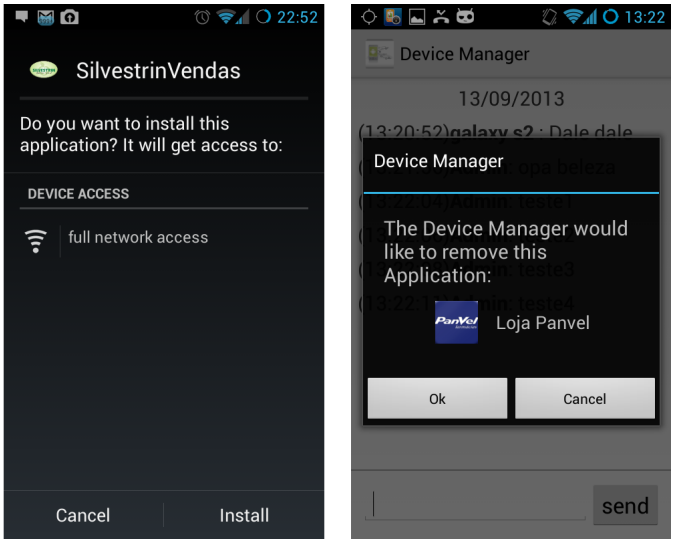

Figura 4.13: Alertas exibidos ao usuário quando um aplicativo é instalado ou removido remotamente.

4.4.2.2 Instalar e Remover Aplicativos

O serviço *DeviceAppsMonitorService* é responsável por tratar as requisições para instalar e remover aplicativos. Para ambas operações é necessário a autorização do usuário do dispositivo. A figura 4.13 mostra os alertas exibidos ao usuário quando um aplicativo é instalado ou removido no dispositivo. A mensagem para instalação de um aplicativo, vinda pelo GCM, possui o URL para o arquivo APK a ser instalado. A mensagem para remover um aplicativo traz no nome no *Package* a ser removido.

Esse serviço também monitora os aplicativos instalados no aparelho e envia ao servidor quando requisitado. O controle da lista de aplicativos funciona da seguinte maneira: o serviço recebe uma notificação sempre que alguma mudança acontecer na lista de aplicativos instalados no aparelho. Quando chega essa notificação o serviço salva um *timestamp* para indicar a última atualização da lista. Então, uma requisição pela lista de aplicativos do aparelho chega contendo o *timestamp* da lista que está no servidor. Logo, se o *timestamp* recebido for diferente do que está salvo então a lista atualizada é enviada. Essa lista pode ter muitos aplicativos e inclui o ícone de cada aplicativo, logo ela deve ser enviada apenas quando realmente for necessário.

4.5 Aplicação Web

A aplicação web (seção 3.4), é o meio por onde o administrador dos dispositivos interage no sistema. Essa aplicação dá ao administrador a capacidade de visualizar o inventário de dispositivos móveis registrados da empresa, trocar mensagens de *chat* com os dispositivos móveis, instalar aplicativos remotamente, entre outras que serão descritas nesta seção.

A aplicação web foi desenvolvida utilizando HTML, JavaScript, Ajax(Asynchronous JavaScript and XML), JQuery. Para a implementação do leiatute das páginas web foi utilizada a ferramenta *TwitterBootstrap* (TWITTER, 2013).

Para facilitar a descrição, a aplicação web é dividida por telas. Para cada tela são explicadas as funcionalidades disponíveis ao administrador e como acontece a interação da aplicação com o servidor para a busca dos dados a serem exibidos. As próximas

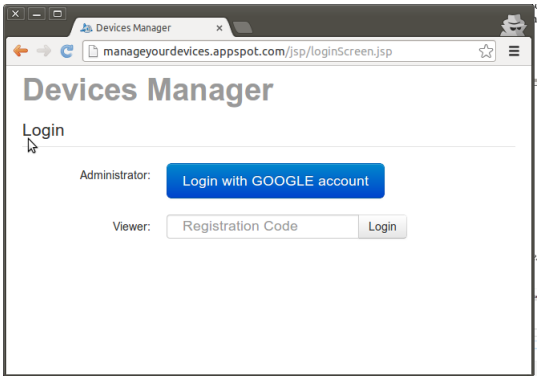

Figura 4.14: Tela de Login do Sistema *Devices Manager*.

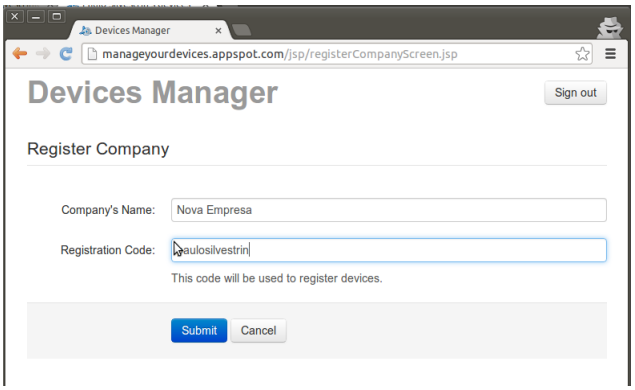

Figura 4.15: Tela de registro de uma nova empresa no sistema *Devices Manager*.

seções descrevem primeiramente as telas de autenticação e registro da empresa, depois é descrita a tela inicial que exibe o inventário de dispositivos Android registrados. Nas seções seguintes é a vez da tela de cadastro e *upload* de aplicativos, depois da tela de detalhes e configuração remota do aparelho. E, por fim, a tela com os dados da empresa e cadastro de novos administradores.

4.5.1 Autenticação do Administrador e Registro de Empresa

A aplicação web utiliza para autenticação o Google Account do usuário Administrador. Foi escolhida essa forma de autenticação pela facilidade de integração com o *Google App Engine*. O sistema utiliza apenas o *e-mail* do administrador para identificar na base de dados quem está entrando no sistema e redirecionar para a empresa a qual ele pertence.

A figura 4.14 mostra a interface de *login*, o administrador ao clicar em *Login with GOOGLE Account* será redirecionado a página padrão de autenticação do Google. Portanto, para o Administrador do sistema se autenticar no *Devices Manager* é obrigatório que ele tenha ou crie uma conta do Google.

Após a autenticação, o Administrador, que ainda não registrou a sua empresa, é redirecionado para a tela de registro como da figura 4.15. Nessa tela, o Administrador preenche os campos com o nome da empresa e o código de registro e submete o formulário. O servidor irá checar se o código de registro escolhido é único, caso contrário, o administrador será avisado a escolher um outro código. Esse código de registro é a identificação da empresa no sistema e também será utilizado para registrar dispositivos para essa empresa. Depois da autenticação e registro da empresa o administrador é redirecionado para a tela

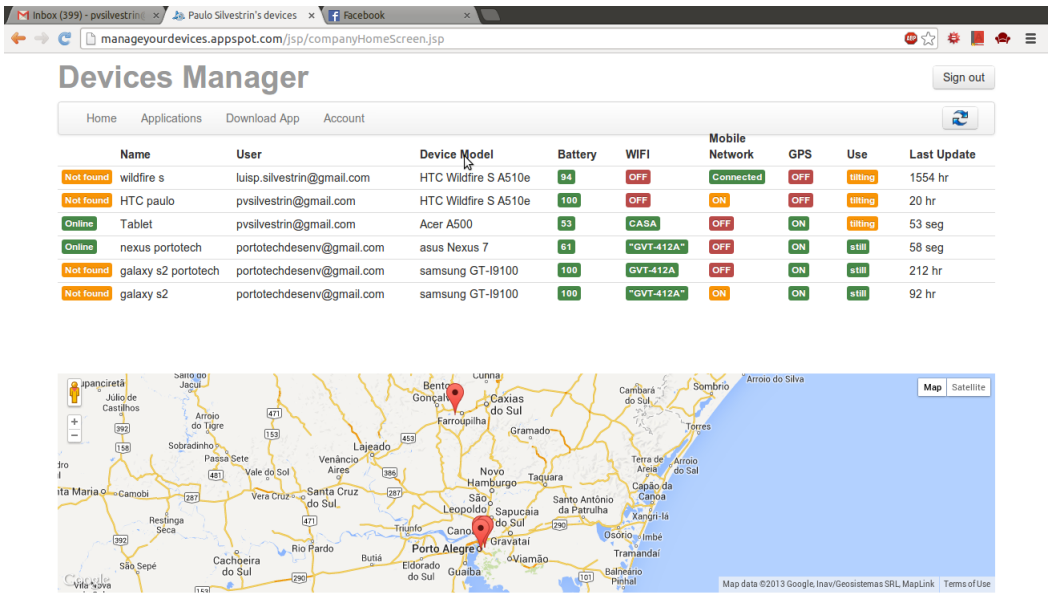

Figura 4.16: Tela inicial do sistema. Mostra o inventário de dispositivos.

inicial do sistema.

4.5.2 Tela Inicial - Inventário de Dispositivos

A tela inicial do *Devices Manager* exibe status geral de todos os dispositivos móveis registrados da empresa a qual o administrador pertence. Como pode ser observado na figura 4.16, a tela exibe a lista com dados de todos os dispositivos e o mapa com a localização deles. A lista mostra, da esquerda para direita, as seguintes informações:

1. Nome do dispositivo;
2. *E-mail* do usuário registrado no dispositivo;
3. Modelo do dispositivo;
4. Nível da bateria;
5. Status da rede *wireless*;
6. Status da rede móvel: 3G ou 4G;
7. Status do GPS;
8. Exibe a situação em que o dispositivo se encontra: parado, inclinado, no carro, andando a pé, na bicicleta;

Abaixo da lista de dispositivos tem o mapa com a disposição deles pela sua localização. O status dos dispositivos e sua localização são atualizados dinamicamente utilizando a tecnologia Ajax. Ou seja, o cliente web faz *polling* no servidor para buscar a lista dos dispositivos com o status atualizado. Desta maneira, sempre que algum dispositivo enviar informações ao servidor a aplicação web irá exibir os dados atualizados com um atraso de, no máximo, a frequência do *polling*. Nessa implementaçao foi utilizado o valor experimental de 4 segundos para frequência do *polling*.

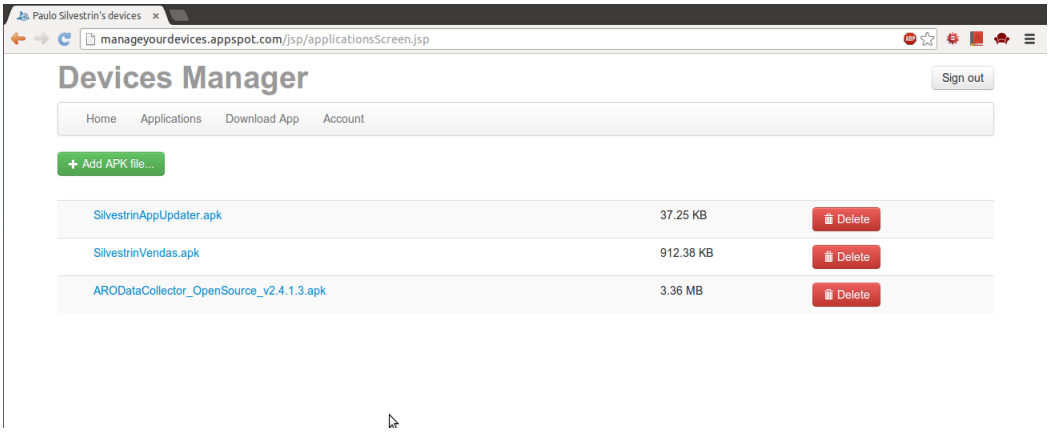

Figura 4.17: Tela de gerenciamento dos aplicativos da Empresa.

No canto superior direito da tela o botão de *Refresh* tem a função de solicitar dados atualizados a todos os dispositivos. Os dispositivos que não responderem com dados atualizados dentro de 30 segundos são considerados como não encontrados.

Ao clicar em algum aparelho da lista, o administrador navega para a tela de detalhes deste dispositivo.

Todas as telas a partir da tela inicial possuem uma barra de navegação com os seguintes itens:

- ***Home*:** navega de volta para a tela inicial;
- ***Applications*:** navega para a tela de *upload* de aplicações(seção 4.5.3);
- ***Dowload*:** tela para fazer download do aplicativo *Device Manager*;
- ***Account*:** tela de detalhes da conta(seção 4.5.5).

4.5.3 Tela Cadastro e *Upload* de Aplicações

A tela de cadastro do aplicativos, figura 4.17, é bastante simples. Apenas permite que o administrados faça o *upload* de arquivos de instalação de aplicativo Android (APKs) para o servidor. O administrador deve clicar em *Add APK File* ou arrastar o arquivo que deseja para dentro da página. Os aquivos carregados são armazenados utilizando a *API Blobstore* (DEVELOPERS, 2013e) oferecida pelo App Engine. Uma referência ao arquivo é salva na coluna *Applications* da tabela *Company*. O *upload* de aplicativos é importante, pois apenas aplicativos que foram previamente cadastrados podem ser instalados remotamente nos dispositivos.

4.5.4 Tela de Configuração do Dispositivo

Nessa tela é onde o administrador dos aplicativos pode visualizar mais dados dos dispositivos, enviar e receber mensagens de *chat* e instalar e remover aplicativos remotamente. Essa tela é dividida em 3 módulos que interagem com o servidor de maneira independente para manter as informações sempre atualizadas.

O primeiro módulo é composto pelos dados do status do dispositivo, os mesmos exibidos na tela inicial adicionando apenas o uso da memória interna e externa. Esses dados são atualizados a cada 4 segundos utilizando Ajax. A aplicação web busca através do *webservice* os dados atuais do aparelho em formato JSON e exibe pro administrador.

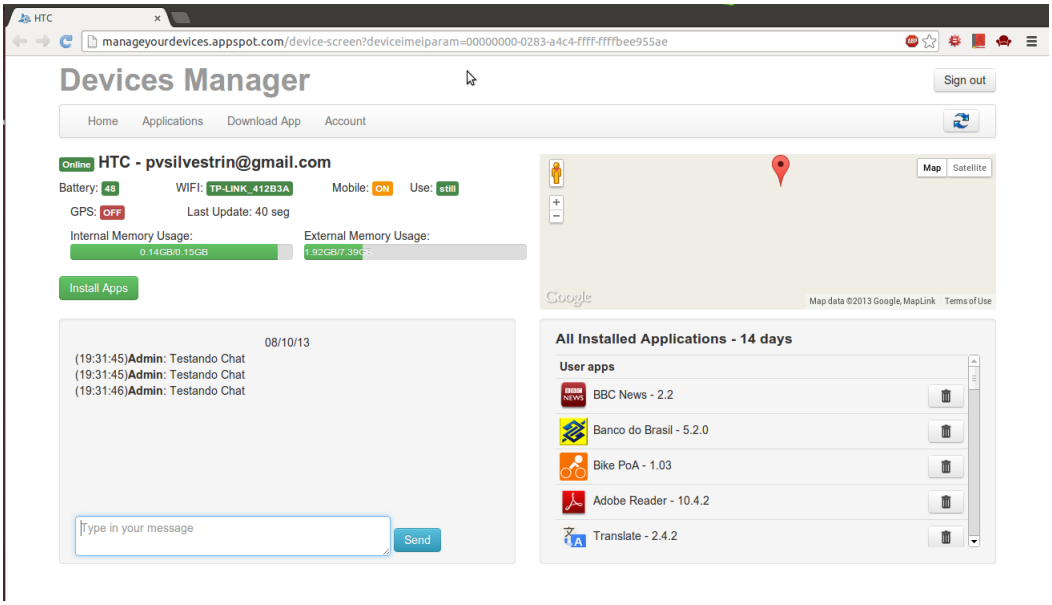

Figura 4.18: Tela de detalhes de um dispositivo.

A área de *chat* é atualizada a cada 2 segundos, para o administrador ter uma experiência de tempo real melhor no caso de um diálogo com o usuário do dispositivo. Nesse caso, a aplicação web a cada 2 segundos, através de um método do *webservice*, envia o *timestamp* da sua última mensagem. Então o servidor confere se existem novas mensagens, caso positivo, responde com a lista das mensagens atualizada. Para enviar menagens de *chat*, a aplicação web utiliza o método do *webservice*, que é responsável por enviar a *push notification* ao dispositivo com a mensagem.

A lista de aplicativos instalados é atualizada de uma maneira semelhante ao *chat*. A aplicação web faz *polling* no servidor a cada 10 segundos mandando o *timestamp* da lista atual dos dispositivos, então apenas no caso da lista estar desatualizada o servidor responde com uma lista atualizada. Para cada aplicativo da lista, que não for um aplicativo do sistema, tem um botão que permite o administrador removê-lo. Ao clicar nesse botão, o servidor manda uma notificação ao dispositivo contendo o nome do pacote do aplicativo a ser removido.

Além das funcionalidades acima citadas, a tela possui um botão no canto superior direito, como na tela inicial, que tem a função de requisitar os dados atualizados do dispositivo. Ou seja, chama o método do *webservice* responsável por mandar a *push notification* ao dispositivo. Nessa tela, esse botão tem uma funcionalidade importante pois a lista de aplicativos instalados no dispositivo nunca é enviada automaticamente pelo aparelho. Ela só é enviada ao servidor quando a requisição é enviada. Portanto, se o administrador quer garantir que a lista de aplicativos está atualizada ele deve clicar neste botão.

4.5.5 Tela de Configurações da Empresa

A tela de configurações da empresa, figura 4.19, mostra os detalhes da empresa a qual o administrador que está usando o sistema pertence. Essa página mostra o nome da empresa, o código de registro, e os *e-mails* dos administradores dos dispositivos da empresa.

Também é possível adicionar novos administradores no campo *Add Administrator*. Esse procedimento é muito simples, é necessário apenas colocar o *e-mail* da conta Goo-

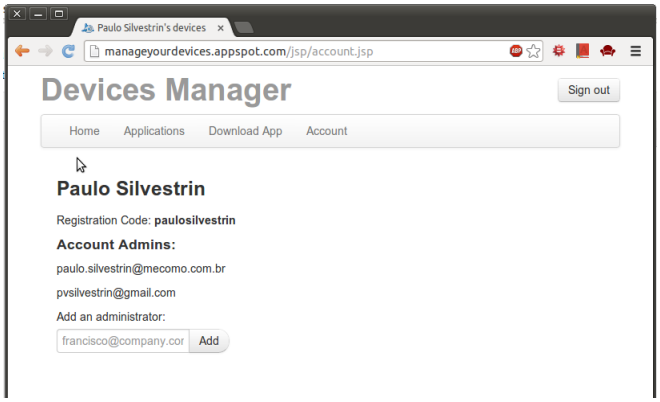

Figura 4.19: Tela com os detalhes da empresa.

gle do novo administrador e clicar no botão *Add*. Assim, quando esse novo administrador entrar no sistema *Devices Manager*, ele vai automaticamente ter acesso a todos os dispositivos da empresa.

4.6 Considerações Finais

O sistema de gerenciamento de dispositivos móveis *Devices Manager* implementado tem como objetivo oferecer as funcionalidades básicas de um MDM usando tecnologias baseadas em código aberto. Com ele, o departamento de TI da empresa tem o controle do inventário e monitoramento dos dispositivos, gerenciamento de aplicativos e suporte técnico via *chat*.

Durante a implementação foi dada uma grande atenção para que o administrador tenha acesso aos dados de uso do dispositivo em tempo real. Ou seja, que o dispositivo envie ao servidor o mais rápido possível quando algum evento importante acontece, como por exemplo, quando a bateria atinge um nível crítico, ou o aparelho se conecta a uma rede *wireless*.

Como Android é um tecnologia ainda recente, muitas atualizações de sistema operacional, bibliotecas e serviços acontecem a todo o momento. Então foi cuidado para utilizar sempre as tecnologias mais recentes disponíveis para que os dados sejam obtidos de maneira eficiente e confiável. Durante a implementação também foi dada a preferência para o uso de *frameworks* e bibliotecas para dar agilidade no desenvolvimento do sistema.

No próximo capítulo, esta implementação será avaliada para verificar se ela atende aos casos de uso básicos de um MDM e quanto ao consumo de dados e de bateria do dispositivo.

5 AVALIAÇÃO

Existem muitas maneiras de avaliar um sistema MDM como, por exemplo, conformidade das funcionalidades, uso de recursos de hardware e dados do dispositivo por parte do módulo cliente, escalabilidade, preocupação com a segurança, etc. Este capítulo descreve a avaliação do sistema *Devices Manager* quanto as suas funcionalidades e quanto ao uso de recursos do dispositivo.

Foram avaliadas a conformidade do sistema implementado quanto as funcionalidades de provisionamento, atualização de software e gerenciamento de falhas que são três das quatro funcionalidades básicas sugeridas pela OMA, como mostrado na seção 2.3. Também foi avaliado o aplicativo móvel quanto ao consumo de bateria e uso de dados.

5.1 Avaliação de Funcionalidades

O sistema oferece os recursos de provisionamento, atualização de software e gerenciamento de falhas através de algumas funcionalidades que são avaliadas nesta seção. A avaliação é feita através de *screenshots* das telas do dispositivo e da interface web de gerência mostrando que as funcionalidades estão de acordo.

5.1.1 Provisionamento

O provisionamento do dispositivo consiste em *download* do aplicativo, instalação e registro do dispositivo. Como pode ser visto na figura 5.1, o *download* do aplicativo pode ser feito através da leitura de um QRCode ou acessando a URL do arquivo. O aplicativo é instalado no dispositivo e executado, na tela inicial são colocados o nome do dispositivo e o código de registro (neste caso *Teste 1* e *teste123*, respectivamente). A figura 5.2 mostra um *screenshot* da tela do dispositivo se registrando e a tela do administrador com o aplicativo registrado.

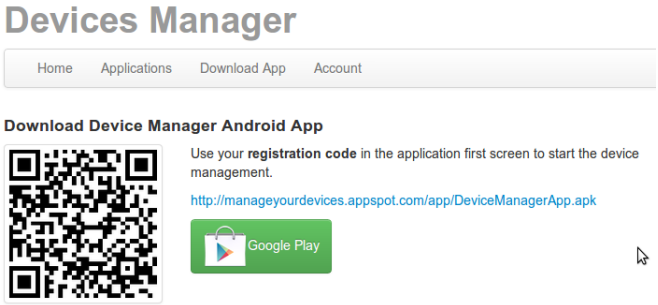

Figura 5.1: Tela de *download* do arquivo de instalação do aplicativo cliente.

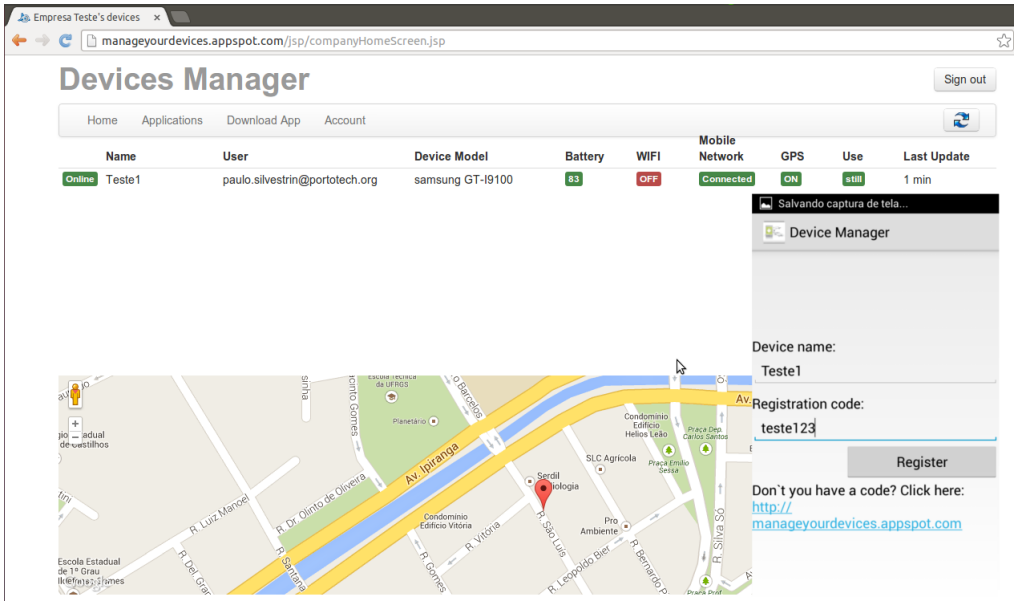

Figura 5.2: Registro do dispositivo.

5.1.2 Atualização de Software

O sistema *Device Manager* disponibiliza que o administrador instale ou remova aplicativos remotamente. A figura 5.3 mostra, na direita, a tela do administrador instalando um aplicativo remotamente e na esquerda a tela do dispositivo pedindo a confirmação da instalação do aplicativo.

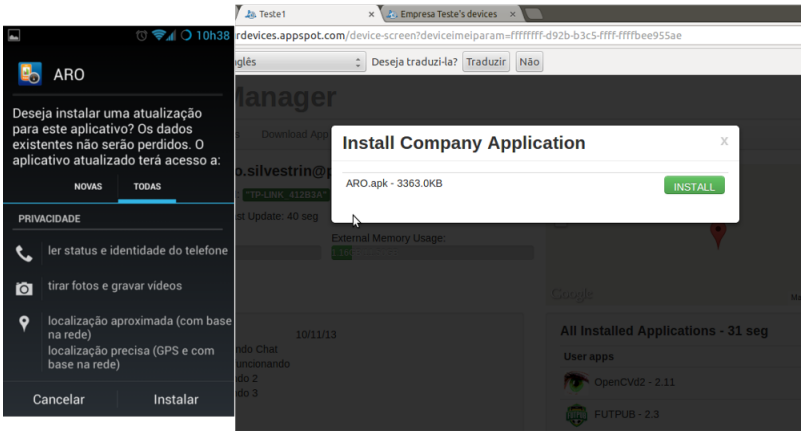

Figura 5.3: Instalação de aplicativo remotamente.

A figura 5.4 mostra, na direita, o Administrador removendo o aplicativo *OpenCVd2* e, na esquerda, a tela do dispositivo recebendo essa notificação.

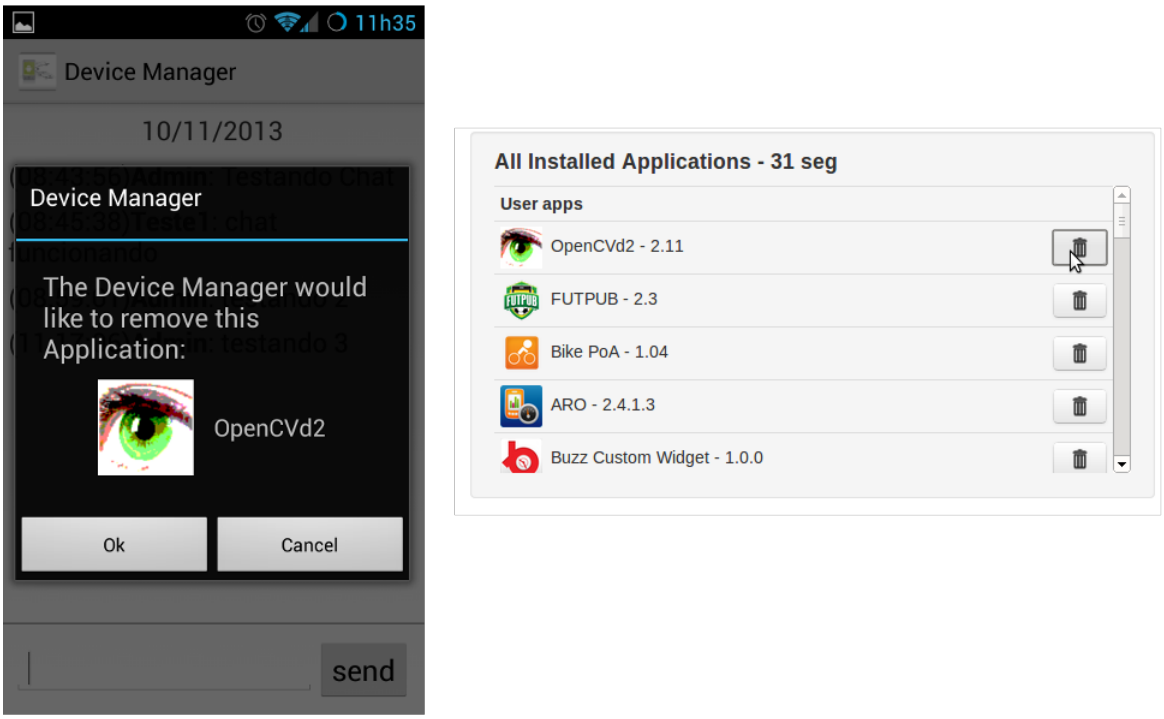

Figura 5.4: Remoção de aplicativo Remotamente.

5.1.3 Gerenciamento de Falhas

O sistema disponibiliza duas maneiras para o administrador lidar com o gerenciamento de falhas: status do dispositivo e *chat*. O monitoramento do status do dispositivo auxilia na detecção de falhas ou mal uso do dispositivo. Já o *chat* é utilizado pelo usuário final para reportar erros, ou falhas, ao administrador.

Para avaliar o monitoramento do status do dispositivo, os dados apresentados pelos sistema (bateria, WIFI, GPS, Rede móvel e uso da memória interna e memória externa) são conferidos com os dados reais do dispositivo disponibilizados no menu de configurações do sistema operacional. A figura 5.5 mostra, na esquerda, os *screenshots* de telas do dispositivo contendo as informações para conferência com os dados apresentados na tela do administrador, na direita.

A figura 5.6 tem o objetivo de mostrar a coerência entre as mensagens de *chat* trocadas entre o administrador e o usuário final. O *screenshot* da esquerda mostra a notificação exibida caso o aplicativo não esteja sendo executando no momento em que uma mensagem é recebida. As outras telas mostram área de *chat* do aplicativo, na esquerda, e da interface do administrador, na direita.

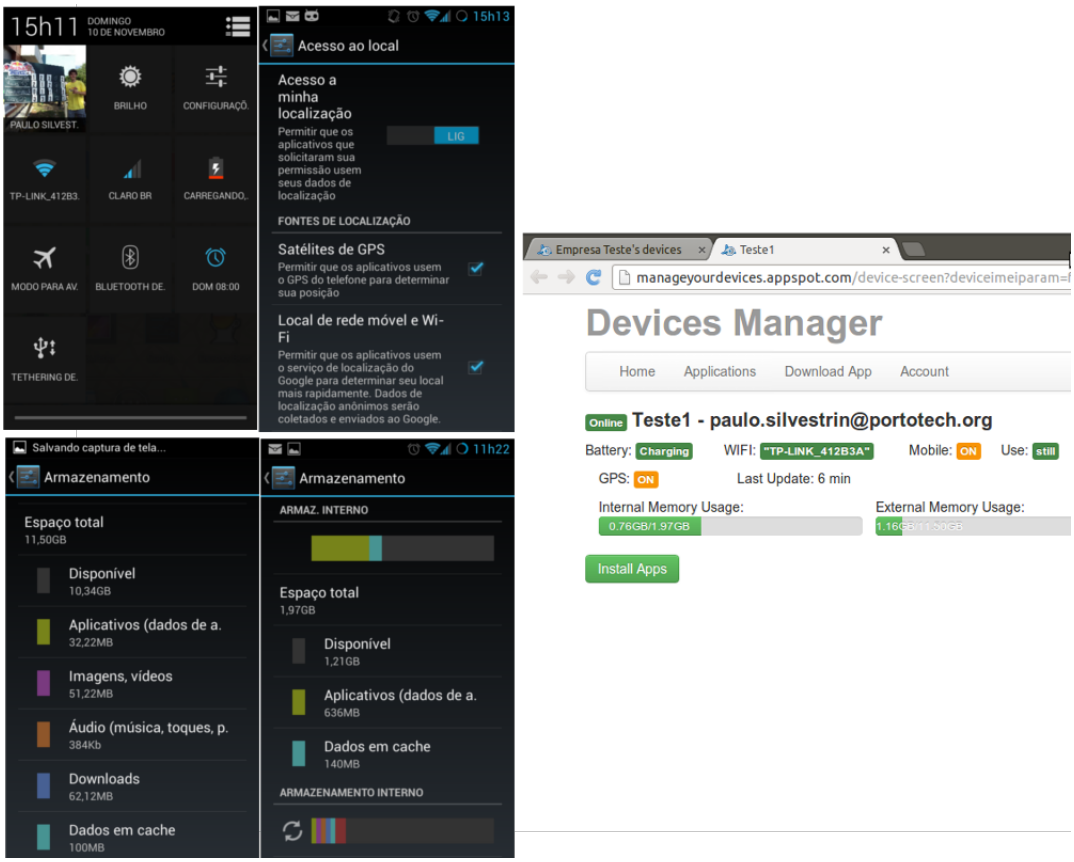

Figura 5.5: Validação dos dados do status do Dispositivo.

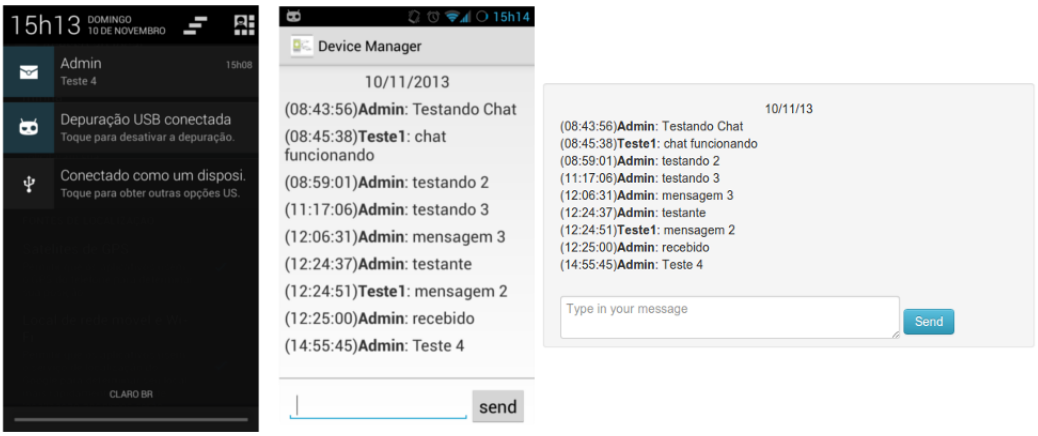

Figura 5.6: Notificação de mensagem de *chat*, tela de *chat* no aplicativo Android e Web

5.2 Avaliação do Consumo

Entre os recursos do dispositivos que um aplicativo consome estão bateria, dados, memória interna e RAM. Como pode ser observado na figura 5.7, o aplicativo Android do *Device Manager* ocupa 2.75 MB de memória interna e 8.3 MB de memória RAM. Existe uma grande quantidade de diferentes dispositivos com sistema operacional Android, e muitos destes dispositivos, mais antigos ou mais econômicos, possuem maiores limitações de hardware, principalmente memória interna e memória RAM. Portanto é importante que o cliente MDM utilize o menos possível esses recursos.

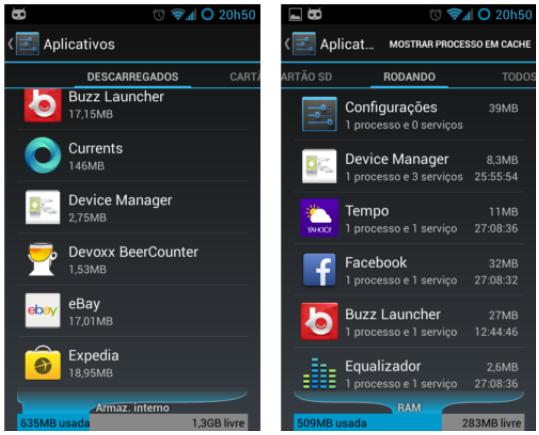

Figura 5.7: Consumo de memória interna e memória RAM do aplicativo *Device Manager*.

A avaliação do consumo de bateria e dados é feita com o auxilio da ferramenta ARO da AT&T (AT&T, 2013). O ARO é composto pelo *ARO Data Collector* e pelo *ARO Data Analyzer*. O *ARO Data Collector* é um aplicativo, que deve ser instalado no dispositivo, usado para capturar o tráfego de dados, informações sobre a rede de dados e do próprio dispositivo. Essas informações são armazenadas em arquivos e, posteriormente, analisadas com base em recomendações e guias de desenvolvimento através do *ARO Data Analyzer*.

Esta ferramenta possui um aplicativo que é instalado do dispositivo, o aplicativo, chamado do *ARO Data Collector* captura o tráfego de dados, informações da rede e do dispositivo, etc. Essas informações são armazenadas em arquivos que serão avaliados com base em recomendações e guias de desenvolvimento utilizando o outro módulo da ferramenta *ARO Data Analyzer*.

Tabela 5.1: Resultados do consumo de bateria e dados do aplicativo.

	Num. Mensagens	Dados	Dados(Total)	Bateria	Bateria(Total)
WiFi	34	63.07Kb	587.72Kb	1104J	1985J
3G	36	60.14Kb	527.56Kb	1536J	2446J

O aplicativo do sistema *Device Manager* foi avaliado quanto a quantidade de dados trafegados pela rede e o consumo da bateria do dispositivo durante 120 minutos em dois cenários diferentes: um com o dispositivo utilizando apenas a rede WiFi e outro com o dispositivo utilizando apenas a rede 3G. Nos dois cenários o dispositivo fica em modo *idle* a maior parte do tempo e o administrador não envia mensagens de *chat* nem faz qualquer requisição.

O resultado desse teste é dado na tabela 5.1. A coluna *Num. Mensagens* indica o número de mensagens com dados atualizados enviadas do dispositivo ao servidor; a coluna *Dados* mostra a quantidade em kbytes de dados trocados entre o aplicativo *Device Manager* e o servidor; a coluna *Dados(Total)* é a quantidade em *kbytes* de dados enviados, ou recebidos, por todos os aplicativos sendo executados no dispositivo durante os 120 minutos de teste; a coluna *Bateria* mostra a quantidade de energia consumida pelo aplicativo *Device Manager*; a coluna *Bateria(Total)* mostra a quantidade de bateria consumida por todos os aplicativos sendo executados durante os 120 minutos de teste. Analisando a tabela 5.1 pode-se concluir que tanto utilizando a rede WiFi, ou a 3G, o aplicativo apresenta o mesmo comportamento quanto ao consumo de dados.

O sistema operacional Android disponibiliza o uso de dados de rede no menu de detalhes do aplicativo. Como pode ser observado na figura 5.8, o aplicativo *Device Manager* utilizou 35.14MB durante 28 dias de uso. Este dado, mesmo não sendo em ambiente controlado e não sabendo de que maneira o aplicativo foi utilizado durante este intervalo de tempo, mostra que o aplicativo não está tendo um consumo abusivo de dados do usuário final.

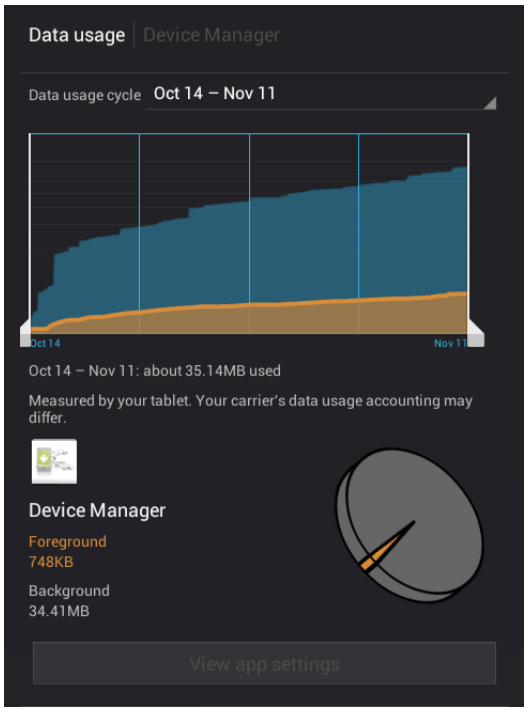

Figura 5.8: Consumo de dados do *Device Manager* durante 1 mês de uso.

5.3 Considerações Finais

A avaliação do sistema *Device Manager* mostrou que o mesmo atende as demandas básicas que um MDM deve ter. Dessa avaliação, também pode-se concluir que o sistema oferece o controle do inventário dos dispositivos e mostra a localização dos mesmos como pode ser observado na figura 5.2.

O resultado do uso de dados e consumo de bateria do aplicativo mostrou que não prejudica, de forma significativa, a autonomia do dispositivo, porém, ainda pode ser melhorado em trabalhos futuros. O uso de dados apresentou-se um pouco elevado dado que

no cenário de teste o dispositivo não sofria muitas alterações, logo a quantidade de mensagens com o status do dispositivo poderia ser muito menor que a obtida. O consumo de bateria está diretamente associado a uso do radio WIFI, ou 3G, diminuindo ainda mais a necessidade de enviar dados ao servidor, automaticamente, irá diminuir o consumo de bateria.

Concluindo, o sistema *Devices Manager* é uma boa ferramenta para empresas que não possuem uma grande quantidade de dispositivos e que desejam monitorar o status e localização dos mesmos. Oferecendo ainda a possibilidade de instalar e remover aplicativos remotamente e *chat* entre usuário final e administrador. Isso tudo a um baixo custo, podendo chegar a zero caso a empresa não passe da cota gratuita de uso do servidor *App Engine*.

6 CONCLUSÃO

O gerenciamento de dispositivos móveis é uma área relativamente nova que surgiu nos últimos anos com o crescimento do poder computacional de smartphones e tablets e com o também crescimento do uso desses dispositivos dentro de empresas e ambientes de trabalho (BYOD - *Bring Your Own Device*).

O desenvolvimento de sistemas que auxiliam nesse gerenciamento, chamados de MDM (*Mobile Device Management*), é bastante desafiador principalmente pelo fato de cada plataforma móvel exigir implementações específicas pela falta de padronização na disponibilização dos recursos e dados necessários para o gerenciamento do dispositivo. O órgão chamado OMA (*Open Mobile Alliance*) visa regulamentar e padronizar recursos relacionados a dispositivos móveis, e tem algumas iniciativas de normatizar funcionalidades relacionadas ao gerenciamento de dispositivos.

Hoje, o mercado para sistemas MDMs é dominado por cinco grandes empresas que oferecem sistemas muito completos com funcionalidades para empresas que desejam ter o total controle e monitoramento de seus dispositivos e documentos. Na pesquisa realizada neste trabalho não foi encontrado nenhuma solução baseada na filosofia de software livre.

O sistema *Devices Manager*, desenvolvido neste trabalho, é um software livre cuja licença que será utilizada ainda não foi definida, para gerenciamento de dispositivos móveis com o objetivo de ter usabilidade simples e de não apresentar impactos no consumo de recursos do dispositivo cliente. Além disso, a ferramenta implementada busca oferecer ao administrador dos dispositivos, em tempo real, uma visão atualizada das informações associadas aos dispositivos.

Para o sistema *Devices Manager* foi implementado um servidor, uma aplicação Android e uma aplicação web. O servidor possui um *webservice*, para a comunicação com as aplicações, um banco de dados e utiliza o GCM *Google Cloud Messaging* para enviar mensagens para a aplicação Android. A aplicação Android tem a funcionalidade de *chat* com o administrador, além disso, coleta dados de uso do dispositivo e envia o servidor. A aplicação móvel implementada oferece ao administrador o controle do inventário, monitoramento e localização dos dispositivos, além de instalar e remover aplicações remotamente e *chat*.

O sistema implementado atendeu ao desafio de proporcionar ao administrador uma experiência de atualização dos dados em tempo real e ao mesmo tempo não causou um grande impacto no aplicativo cliente quanto ao uso de dados e bateria do dispositivo. As funcionalidades básica de provisionamento, atualização de software e gerenciamento de falhas foram testadas e avaliadas mostrando que o sistema implementado é uma prova de conceito viável e está pronto para ser utilizado.

Os sistemas MDM oferecem funcionalidades que podem invadir a privacidade do usuário dos dispositivos. O sistema implementado, por exemplo, oferece ao adminis-

trador a localização dos dispositivos e a possibilidade de instalar e remover aplicativos remotamente. Portanto, o uso destas funcionalidades deve ser feito dentro de politicas bem definidas para que a empresa não tenha problemas legais.

A especificação do sistema proposto neste trabalho descreve todas as funcionalidades e recursos desejáveis ao *Devices Manager*. No entanto, como prova de conceito, optou-se implementar um subconjunto dessas funcionalidades. Logo, as funcionalidades especificadas, e não implementadas, são sugestões para uma sequência deste trabalho, a saber:

1. Implementar no cliente Android a funcionalidade de gerenciamento de aplicativos que possibilita o usuário final instalar e atualizar aplicações da empresa.

2. Na central de gerenciamento web é importante a adição de recursos de configuração por onde o administrador possa habilitar ou desabilidar GPS, *Bluetooth*, *wireless*. Implementação do armazenamento do histórico do status e localização do dispositivos e a visualização deste histórico na tela de detalhes do dispositivo.

3. Implementar uma forma de buscar o histórico das mensagens de chat no aplicativo. Pois da maneira que foi implementado, quando o aplicativo é desinstalado, ou os dados da aplicação são removidos, os o histórico de mensagens enviadas e recebidas é perdido. Nenhum método foi implementado para buscar mensagens anteriores no servidor como acontece no *Facebook*.

Além dessas funcionalidades, sugere-se como melhorias para o sistema, otimizações no cliente Android quanto aos serviços que enviam o status do dispositivo visando reduzir o tráfego de dados e consumo de bateria. A interface web do administrador pode ser melhorada com a utilização de *websockets* ao invés de *Polling* para a atualização dinâmica dos dados.

O compartilhamento de arquivos, onde o administrador pode enviar arquivos remotamente ao dispositivo, é uma funcionalidade que pode ser adicionada ao sistema e não foi prevista na especificação. Essa funcionalidade pode ser facilmente adicionada ao sistema atual, sendo implementada de maneira parecida a instalação remota de aplicativos. Também pode-se adicionar ao sistema funcionalidades relacionadas a segurança do dispositivo como bloquear ou limpar os dados do dispositivo remotamente em caso de perda ou furto do aparelho. Porém, essas funcionalidades são mais complexas de implementar pois exigem que o aplicativo tenha acesso a privilégios de administrador do sistema operacional.

REFERÊNCIAS

AIRWATCH. **AirWatch Mobile Device Management**. Disponível em: <http://www.air-watch.com/solutions/mobile-device-management>. Acesso em: agosto 2013.

ALLIANCE, O. M. **Device Management Requirements - Approved Version 1.2**. [S.l.]: 2007 Open Mobile Alliance Ltd. All Rights Reserved., 2007. Disponível em: <http://technical.openmobilealliance.org/Technical/release_program/docs/DM/V1_2_1-20080617-A/OMA-RD-DM-V1_2-20070209-A.pdf> Acesso em: outubro 2013.

AT&T. **AT&T Application Resource Optimizer**. Disponível em: <http://developer.att.com/developer/forward.jsp?passedItemId=13800072>. Acesso em: novembro 2013.

BLOG, G. D. **Google App Engine Blog**: happy birthday high replication datastore: 1 year, 100,000 apps, 0% downtime. Disponível em: <http://googleappengine.blogspot.com.br/2012/01/happy-birthday-high-replication.>. Acesso em: outubro 2013.

DEVELOPERS, A. **Android Intent Reference**. Disponível em: <http://developer.android.com/reference/android/content/Intent.html>. Acesso em: outubro 2013.

DEVELOPERS, A. **Android BroadcastReceiver Reference**. Disponível em: <http://developer.android.com/reference/android/content/BroadcastReceiver.html>. Acesso em: setembro 2013.

DEVELOPERS, A. **Blobstore Java API Overview**. Disponível em: <https://developers.google.com/appengine/docs/java/blobstore/>. Acesso em: outubro 2013.

DEVELOPERS. **Saving Data in SQL Databases**. Disponível em: <http://developer.android.com/training/basics/data-storage/databases.html>. Acesso em: novembro 2013.

DEVELOPERS, S. **How to retrieve the Device Unique ID from android device**. Disponível em: <http://developer.samsung.com/android/technical-docs/How-to-retrieve-the-Device-Unique-ID-from-android-device/>. Acesso em: abril 2013.

FIBERLINK. **Maas360**. Disponível em: <http://www.maas360.com/>. Acesso em: novembro 2013.

GARTNER. **Gartner - Magic Quadrant for Mobile Device Management Software**. Disponível em: <http://www.gartner.com/technology/reprints.do?id=1-1FRIMH0&ct=130523&st=sb>. Acesso em: outubro 2013.

GCM. **Google Cloud Message - Android**. Disponível em: <http://developer.android.com/google/gcm/gcm.html>. Acesso em: agosto 2013.

GOOGLE. **App Engine Service Level Agreement - Google App Engine - Google Code**. Disponível em: <https://developers.google.com/appengine/sla?csw=1/>. Acesso em: agosto 2013.

GSMA. **IMEI Allocation and Approval Guidelines**. Disponível em: <http://www.gsma.com/newsroom/ts-06-6-0-imei-allocation-and-approval-guidelines/>. Acesso em: abril 2013.

IT, C. of. **Thrive on Consumerization**. [S.l.]: Intel vPro Technology, 2010. <http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/vpro-thrive-on-consumerization-of-it-paper.pdf>.

LOCATION. **Location Request - Android**. Disponível em: <http://developer.android.com/reference/com/google/android/gms/location/LocationRequest.html>. Acesso em: setembro 2013.

MATHIAS, C. J. **Choosing an MDM system**: fundamental features for success. Disponível em: <http://searchconsumerization.techtarget.com/tip/Choosing-an-MDM-system-Fundamental-features-for-success>. Acesso em: novembro 2013.

MEMCACHE. **Memcache Java API Overview**. Disponível em: <https://developers.google.com/appengine/docs/java/memcache/>. Acesso em: novembro 2013.

MIKE OLIVER, S. i. **Mobile Device Management for Dummies**. First Edition.ed. England: John Wiley & Sons, Ltd, 2008.

MILETTE, G.; STROUD, A. **Professional Android Sensor Programming**. 1rd.ed. Indianapolis, Indiana: John Wiley e Sons, Inc, 2012.

MOBILEIRON. **MobileIron**. Disponível em: <http://www.mobileiron.com/>. Acesso em: novembro 2013.

MOBILEIRON. **MobileIron Deployment Options**. Disponível em: <http://www.mobileiron.com/en/products/deployment-options>. Acesso em: novembro 2013.

OBJECTFY. **Objectfy-appengine**. Disponível em: <https://code.google.com/p/objectify-appengine/>. Acesso em: agosto 2013.

OMA. **OMA - Open Mobile Alliance**. Disponível em: <http://openmobilealliance.org/>. Acesso em: outubro 2013.

ORACLE. **Java Servlet Technology Overview**. Disponível em: <http://www.oracle.com/technetwork/java/javaee/servlet/index.html>. Acesso em: agosto 2013.

ORACLE. **JavaServer Pages Technology**. Disponível em: <http://docs.oracle.com/javaee/1.4/tutorial/doc/>. Acesso em: setembro 2013.

OVERAIR. **Over-the-air programming**. Disponível em: <http://en.wikipedia.org/wiki/Over-the-air_programming/>. Acesso em: outubro 2013.

PHIFER, L. **Mobile device management checklist**. Disponível em: <http://searchconsumerization.techtarget.com/tip/Mobile-device-management-checklist>. Acesso em: outubro 2013.

POLLING. **Polling**. Disponível em: <http://en.wikipedia.org/wiki/Polling_(computer_science)>. Acesso em: out 2013.

PUSH. **Push technology**. Disponível em: <http://en.wikipedia.org/wiki/Push_technology>. Acesso em: out 2013.

RECOGNITION. **Activity Recognition Client - Android**. Disponível em: <http://developer.android.com/reference/com/google/android/gms/location/ActivityRecognitionClient.html>. Acesso em: outubro 2013.

RICHARDSON, L.; RUBY, S. **RESTful Web Services**. First Edition.ed. United States of America: OReilly Media, 2007.

TIA, T. I. A. **MOBILE EQUIPMENT IDENTIFIERS (MEID)**. Disponível em: <http://www.tiaonline.org/standards/numbering-resources/mobile-equipment-identifiers-meid>. Acesso em: abril 2013.

TIA, T. I. A. **ELECTRONIC SERIAL NUMBERS (ESN) AND MEID**. Disponível em: <http://www.tiaonline.org/standards/numbering-resources/electronic-serial-numbers-esn-and-meid>. Acesso em: junho 2013.

TWITTER. **Bootstrap**. Disponível em: <http://getbootstrap.com/2.3.2/>. Acesso em: setembro 2013.

WORLD, C. **MDM tools**: features and functions compared. Disponível em: <http://www.computerworld.com/s/article/9238981/MDM_tools_Features_and_functions_compared>. Acesso em: novembro 2013.