CARLOS ALBERTO FIGUEIREDO

ANÁLISE E IMPLEMENTAÇÃO DE REGRAS SENSÍVEIS AO CONTEXTO PARA PROTEÇÃO E CONTROLE DE DISPOSITIVOS MÓVEIS COM SISTEMA OPERACIONAL ANDROID

Monografia apresentada ao PECE Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof. Dr. Kechi Hirama

São Paulo
2012

FICHA CATALOGRÁFICA

Figueiredo, Carlos Alberto
Análise e implementação de regras sensíveis ao contexto para proteção e controle de dispositivos móveis com sistema operacional Android / C.A. Figueiredo. -- São Paulo, 2012.
92 p.

Monografia (MBA em Tecnologia de Software) – Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.

1. Sistemas operacionais 2. Computação móvel I. Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia II. t.

DEDICATÓRIA

Dedico este trabalho à minha esposa

AGRADECIMENTOS

A minha esposa, Valéria, sem cuja perseverança e companheirismo, esta realização não seria possível.

Ao meu sonho de aprender e ser um ser humano melhor.

Ao Professor Dr. Kechi Hirama pela orientação e exemplo, e por acreditar que eu conseguiria, sempre. E a todos que direta ou indiretamente colaboraram na execução deste trabalho.

RESUMO

A proteção e o controle de uso de dispositivos móveis através de regras sensíveis ao contexto é uma opção estudada nos dias de hoje para possibilitar uma adequada experiência do usuário na utilização destes dispositivos.

Ao utilizar regras sensíveis ao contexto para a proteção e controle de utilização o consumo de energia de bateria do dispositivo sofre um aumento de consumo dependendo da técnica e implementação adotada.

Como um sistema operacional aberto o Android tem se consolidado como tendência de uso para dispositivos móveis no mercado.

Em função disso os softwares maliciosos para o Android têm crescido em número e complexidade.

Neste trabalho propõe-se fazer uma comparação entre os resultados obtidos na literatura e um experimento para levantar possíveis pontos de atenção para obter uma adequada experiência do usuário. Para isto, são abordados os conceitos básicos do mecanismo de proteção padrão do sistema operacional Android e as técnicas alternativas para adicionar proteção e controle de uso, em particular por regras sensíveis ao contexto.

ABSTRACT

The protection and control of mobile devices through context-sensitive rules is an option studied today to allow an adequate user experience in using these devices.

When using context-sensitive rules for the protection and utilization control power consumption of the device battery consumption is increased depending on the technique adopted and implementation.

As an open operating system Android has been established as a trend to use mobile devices on the market.

Because of this malicious software to the Android have grown in number and complexity.

This work proposes to make a comparison between the results obtained in the literature and an experiment to raise possible points of attention for a proper user experience. For this, the article discusses the basic concepts of the mechanism of protection standard operating system Android and alternative techniques to add protection and usage control, in particular for context-sensitive rules.

LISTA DE ILUSTRAÇÕES

Pág.

LISTA DE TABELAS

Pág.

LISTA DE ABREVIATURAS E SIGLAS

IDS - Intrusion Detection System

NFC - Near Field Communication

JVM - Java Virtual Machine

SDK - Software Development Kit

NDK - Native Development Kit

IDE - Integrated Development Environment

API - Application Programming Interface

POSIX - Portable Operationg System Interface

JNI - Java Native Interface

CVE - Common Vulnerabilities and Exposures

CPU - Central Processing Unit

RAM - Random Access Memory

SMS - Short Message Service

MMS - Multimedia Messaging Service

VPN - Virtual Privite Network

GPS - Global Positioning System

DNS - Domain Name System

CAAC - Context Aware Access Control

SUMÁRIO

Pág

1. INTRODUÇÃO

As áreas como a Medicina, Engenharia e Economia são apoiadas por sistemas tecnológicos que exigem aperfeiçoamentos e superação permanentes. Atualmente, os dispositivos móveis tais como celulares, *tablets* e *notebooks* estão muito difundidos com o avanço da Internet. Este capítulo apresenta as motivações, o objetivo, as justificativas e a estrutura deste trabalho.

1.1 Motivações

Reflexo e conseqüência da incorporação dos inúmeros serviços integrados à Internet, que vêem sendo oferecidos basicamente em todos os segmentos do mercado, a vulnerabilidade à danos causados por sofwares maliciosos (Malwares), seguem crescendo exponencialmente nos sistemas operacionais de dispositivos móveis (LEAVIT, 2011), (STUECKLE, 2011), (SHASBTAI, et al 2010).

A computação móvel introduziu uma diversidade enorme de serviços, tais como acesso às personalizações, opções para pagamentos de contas e muitos outros recursos, e isso, conseqüentemente, tornou-a cada vez mais vulnerável a ataques à sua proteção (LEAVIT, 2011), (STUECKLE, 2011), (SHASBTAI, et al 2010).

Devido à rápida e ascendente popularização do uso de serviços via Internet, a questão da proteção de acesso (ou proteção) para estes dispositivos, vem conquistando grande relevância não só junto aos fabricantes de softwares, mas também para empresas prestadoras de serviços e usuários (LEAVIT, 2011), (STUECKLE, 2011), (SHASBTAI, et al 2010).

As vendas de dispositivos móveis cresceram 96% em relação ao 3º trimestre de 2009, representando 19,3% do total de vendas de celulares naquele ano.

A entidade Gartner Group dos EUA estima que as vendas mundiais de telefones móveis, no 3º trimestre de 2010, totalizaram 417 milhões de aparelhos, dos quais 81 milhões são telefones celulares inteligentes (GARTNER, 2011).

O sistema operacional Android da empresa Google dos EUA, está presente em 25,5% das vendas de dispositivos móveis e cresceu 3,5% em relação a 2009. Apontado na pesquisa mais recente da empresa Gartner (2011), mais da metade dos smartphones vendidos no mundo, no terceiro trimestre de 2011, são equipados com Android, e foram vendidos cerca de 60,5 milhões de telefones, o que supõe que esse sistema operacional possui 52,5% dos usuários contra 25,3% no mesmo período de 2010.

Como um sistema operacional aberto (baseado no Kernel 2.6 do Linux), com um *middleware* e um conjunto de aplicações chave, o Android tem se consolidado como tendência de uso para dispositivos móveis no mercado. Tanto é, que em apenas dois anos após seu lançamento, já era o segundo sistema operacional mais usado em dispositivos móveis em todo o mundo, ficando atrás apenas do sistema operacional Symbian, da empresa Nokia da Finlândia (LEAVIT, 2011), (STUECKLE, 2011), (SHASBTAI, et al 2010).

Segundo outra pesquisa divulgada pela empresa Canalys dos EUA (2011), no 2º trimestre de 2011, o Android tornou-se líder mundial em dispositivos móveis. A pesquisa analisou o mercado de dispositivos móveis em 56 países e concluiu que o Android lidera em 35 deles, ou seja, detém 48% do mercado global no segundo trimestre.

De acordo com a empresa CNet dos EUA (2011), os lançamentos baseados no Android no segundo trimestre de 2011, ultrapassaram os 51,9 milhões, o que representa um crescimento de 379% se comparado ao mesmo período do ano

de 2010.

Ainda de acordo com essa pesquisa, foram entregues 107,7 milhões de unidades de dispositivos móveis ao redor do mundo, representando um aumento de 73%, ano após ano. Desse montante, 39,8 milhões foram lançados na Ásia e 35 milhões de unidades na Europa e África. No continente americano, foram lançados 32,9 milhões de aparelhos dispositivos móveis.

Segundo a empresa TG Daily dos EUA (2011), no mesmo período o sistema operacional iOS da empresa Apple dos EUA ultrapassou o Symbian da empresa Nokia e garantiu o segundo lugar com 19% do mercado. A Apple tornou-se, portanto, a líder individual em vendas de dispositivos móveis, tirando a liderança da empresa Nokia.

Da mesma proporção, os softwares maliciosos para o Android têm crescido em número e complexidade. A mais recente ameaça descoberta, por exemplo, é capaz de gravar todas as conversas telefônicas das vítimas. Um relatório divulgado pela companhia de proteção móvel Lookout Mobile Security dos EUA mostraram que (em agosto de 2011), usuários desta plataforma têm 2,5 vezes mais chances de baixar um software malicioso do que há seis meses atrás (MYLOOKOUT, 2011).

De acordo com Shasbtai, et al (2010), dois anos de evolução de vírus de dispositivos móveis são equivalentes a 20 anos de trabalho em vírus de computadores convencionais.

A Lookout Mobile Security estima que somente na primeira metade do ano, meio milhão de usuários de dispositivos com Android sofreram com alguma contaminação maliciosa. Os softwares maliciosos para dispositivos móveis evoluem rapidamente, pelo fato de que os desenvolvedores de vírus estão constantemente adquirindo experiência em computadores convencionais e na Internet mundial. Também notou-se, que os cibercriminosos por trás do desenvolvimento desses malwares, estão aprimorando suas técnicas. Agora, além de enviar softwares maliciosos para o Android Market (Loja virtual oficial de aplicações da Google para o Android), para enganar as vítimas, eles

chegam a comprar espaços publicitários virtuais, onde induzem o usuário a baixar aplicações aparentemente legítimas (como updates, por exemplo), mas que na verdade o direcionam para códigos nocivos (MYLOOKOUT, 2011).

Até agora, os principais surtos foram uma pandemia em escala limitada. No entanto, uma vez que a quota de mercado de dispositivos móveis tende a aumentar significativamente nos próximos anos (estima-se quase cinco vezes até 2013), os dispositivos móveis se tornarão terreno fértil para todo tipo de vírus. Outro grande fator de atração criminosa é o fato de que os dispositivos móveis são freqüentemente usados para fins comerciais e são susceptíveis por conter informações sigilosas. Eles ainda promovem acesso remoto a uma companhia e seus dados mais sensíveis, o que pode levar ao vazamento de informações, caso estes dispositivos móveis sejam invadidos (SHASBTAI, et al 2010).

Com isso, o desafio para garantir a proteção nos dispositivos móveis com Android está se tornando muito semelhante às necessidades de um computador pessoal convencional (SHASBTAI, et al 2010).

Empresas de todas as partes do mundo já anunciaram sua intenção de modificar suas soluções de proteção para dispositivos com o Android. Dentre as soluções, estão inclusos diferentes softwares antivírus e sistemas de detecção de intrusão (*Intrusion Detection System* - IDS) (inclusive alguns já usados na computação convencional) (STUECKLE, 2011), (SHASBTAI, et al 2010), (BAI, et al 2010), (ENCK, OCTEAU, MCDANIEL, CHAUDHURI, 2010).

Uma série de mecanismos de proteção (Definimos para este trabalho com proteção as técnicas que proporcionam que as informações, o dispositivo físico e o controle de uso estejam adequadas para a utilização do usuário) já está disponível na plataforma padrão do Android, como Identificadores de usuários, Label de permissão e a Assinatura das aplicações. No entanto, o modelo de controle de permissão do Android é pouco refinado e incompleto. Quando uma aplicação apresenta uma lista de permissões na sua instalação (Arquivo AndroidManifest. xml), por exemplo, o usuário só pode optar por todas ou por nenhuma destas permissões. Não há múltiplas escolhas (SHASBTAI, et al

2010).

Além disso, o usuário não pode revogar ou alterar as permissões cedidas de uma aplicação, exceto se esta aplicação for reinstalada. Não há mecanismos para que o usuário possa impor restrições sensíveis ao contexto de uso, em dados e recursos do Android.

Ao ativarmos regras sensíveis ao contexto (Definimos para este trabalho como sensíveis ao contexto, se estas regras usam o contexto e fornecem informações relevantes e / ou serviços para o usuário, onde relevância depende da tarefa do usuário) para proteção e ou controle de utilização de dispositivos móveis baseados em Android, dependendo do consumo de recursos do dispositivo (GPS e 3G por exemplo), a quantidade de horas disponíveis de energia na bateria pode ser comprometida e acarretar em uma indisponibilidade que afeta negativamente a experiência do usuário. Se o usuário precisa do dispositivo protegido por 10 ou 14 horas dependendo de como esta proteção for implementada, isto pode ser comprometido (CONTI, NGUYEN, CRISPO, 2010).

1.2 Objetivo

O objetivo do trabalho é analisar e implementar regras sensíveis ao contexto para proteção e controle de dispositivos móveis, com o sistema operacional Android com intuito de melhorar a experiência do usuário.

Para que o usuário tenha uma experiência de utilização do dispositivo móvel da forma desejada, regras de identificação de sua localização geográfica e dos contextos associados a estas posições devem ser implementadas. A ressalva é que o gerenciamento da energia consumida pela bateria do dispositivo é o ponto critico para possibilitar esta experiência de maneira adequada (SHASBTAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010), (ENCK, OCTEAU, MCDANIEL, CHAUDHURI, 2010).

1.3 Justificativas

O primeiro vírus de dispositivos móveis denominado Cabir, foi lançado em 2004 pelo grupo de criação do vírus 29ª, como prova conceitual de um vírus auto-replicante (SHASBTAI, et al 2010).

Desde então, várias centenas de vírus de dispositivos móveis vem surgido, muitos dos quais contendo códigos maliciosos e causando vários níveis de danos (SHASBTAI, et al 2010).

O número crescente de ataques às plataformas móveis, juntamente com o uso cada vez maior de proteção, vem levando muitos vendedores e pesquisadores a proporem uma variedade de soluções de proteção adicional para as plataformas móveis (LEAVIT, 2011), (LI, IM, 2011).

Na análise da literatura acadêmica relacionada ao trabalho, observa-se que as pesquisas existentes, referente à proteção e/ou adição de proteção aos dispositivos móveis tem se concentrado em sistemas de detecção de intrusão (IDS) (LEAVIT, 2011), (STUECKLE, 2011), (SHASBTAI, et al 2010).

De acordo com Shabtai (2010), técnicas de proteção adicionadas a camada padrão de proteção do Android resolvem falhas de proteção desta. As técnicas de controle de utilização de dispositivos móveis por regras sensíveis ao contexto (*Context Aware Computing*) foram mencionadas na literatura pesquisada, mas não detalhadas em combinação com as demais técnicas de proteção.

Os trabalhos de Conti, Nguyen e Crispo (2010) e Bai, et al (2010) descrevem duas técnicas de controle de utilização de dispositivos móveis por regras sensíveis ao contexto em detalhe, mas não em combinação com outras técnicas de proteção e gestão de energia da bateria em função da utilização das mesmas.

Este trabalho apresenta um experimento para verificar a influência dessas técnicas e uma comparação com os resultados obtidos em Conti, Nguyen e

Crispo (2010). A partir da análise desta comparação sugere pontos de atenção em relação à utilização dessas técnicas.

1.4 Estrutura do Trabalho

O Capítulo 1 INTRODUÇÃO apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

O Capítulo 2 DESCRIÇÃO DOS CONCEITOS apresenta o sistema operacional Android, o seu mecanismo padrão de proteção e o estado atual dos mecanismos de proteção adicionais apresentando uma análise das técnicas e mecanismos para proteção à sua camada padrão. Os conceitos de regras sensíveis ao contexto, gestão de energia e a definição de melhor experiência do usuário também são definidos.

O Capítulo 3 ANÁLISE DO PROBLEMA DE PROTEÇÃO E CONTROLE DE USO, descreve uma análise de como as menores perdas de desempenho e menor consumo possível de energia de bateria com diferentes níveis de proteção por *Context-Aware Computing* (proteção de controle de uso por regras sensíveis ao contexto) no Android.

O Capítulo 4 PROPOSTA DE EXPERIMENTO E RESULTADOS descreve a analise comparativa de sobrecarga nos níveis de energia e o resultado da utilizar o mecanismo de proteção por regra sensíveis ao contexto, adicionados a camada de proteção padrão da sistema operacional Android.

O Capítulo 5 CONSIDERAÇÕES FINAIS descreve e coloca em destaque a análise realizada neste trabalho, com foco no impacto da adição da camada de proteção (Básica mais a ativação de regras sensíveis ao contexto) e a gestão de energia de bateria na plataforma o Android. Sugestões para futuros trabalhos.

2 DESCRIÇÃO DOS CONCEITOS

Neste capítulo são descritos os conceitos envolvidos neste trabalho enfatizando a arquitetura do sistema operacional Android, a sua proteção padrão, as técnicas de proteção adicional, a proteção ativada por regras sensíveis ao contexto e a importância da gestão adequada do consumo de energia da bateria dos dispositivos móveis. Descreve-se também uma noção de uma melhor experiência do usuário na utilização de seu dispositivo móvel.

2.1 Sistema Operacional Android

O sistema operacional Android estreou nos dispositivos móveis G1, da empresa HTC, em outubro de 2008. Neste dispositivo não havia recursos multitoque nem teclado virtual em sua versão 1.0/1.1. Em seguida, foi anunciada em 2009 a versão Cupcake 1.5 que trouxe a gravação e a exibição de vídeos, além de teclado virtual. Teve a interface com o usuário aperfeiçoada e iniciou os nomes sugestivos associados a doces para as versões do Android. Em setembro de 2009, a versão Donut 1.6 foi lançada, a caixa de busca permitia pesquisas nos dispositivos e ficou fácil criar aplicações para diferentes formatos de tela. A versão Eclair 2.0/2.1 foi anunciada um mês depois e inaugurou o suporte a múltiplas contas para sincronizar e-mails e contatos. Em maio de 2010 foi anunciada a versão Froyo 2.2, mais veloz, trouxe suporte ao aplicativo Flash, que permitiu a gravação de aplicações no cartão e liberou o uso do aparelho como hotspot (Ancoragem do dispositivo móvel com um computador para se comportar como roteador 3G). Foi em dezembro de 2010 que aversão Gingerbread 2.3 chegou, a velocidade melhorou, ajudando o desempenho de jogos e inaugurou o suporte à tecnologia NFC (Near Field Communication), que permite utilizar o dispositivo móvel para realizar pagamentos. Liberada em fevereiro de 2011 a versão Honeycomp 3.0, trouxe um design mais bonito, voltado para tablets, mas seus defeitos só foram corrigidos na versão 3.1. No mês de outubro de 2011, foi anunciada a versão Ice Cream Sandwich 4.0, que unificou as interfaces de tablets e outros dispositivos móveis (smartphones e celulares), tornado-as mais simples e

intuitivas (DEVELOPER ANDROID, 2011).

A arquitetura do Android é a mesma para todas as versões citadas acima. Os problemas de proteção e aspectos gerais de proteção do sistema operacional Android são válidos independente da sua versão (DEVELOPER ANDROID, 2011).

Para entender o sistema operacional Android descrevem-se a seguir a sua arquitetura em camadas, sendo desde o hardware (camada inferior) até as aplicações (camada superior) (LEAVIT, 2011), (SHASBTAI, et al 2010).

A arquitetura do Android é apresentada na Figura 1, particularmente observando suas característica de proteção (LEAVIT, 2011), (SHASBTAI, et al 2010).

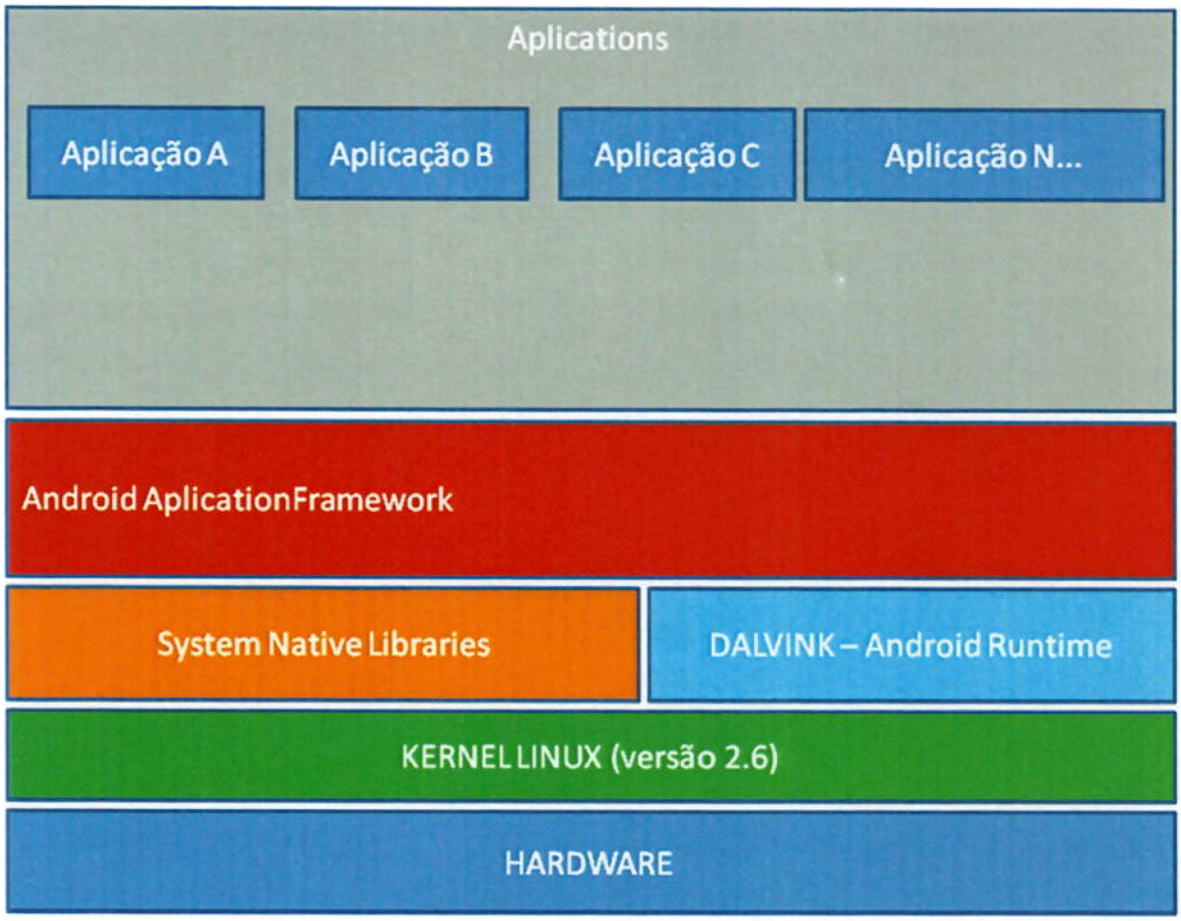


Figura 1. Arquitetura e mecanismo de proteção padrão do sistema operacional Android

O Android é um ambiente de execução de aplicações para dispositivos móveis, que inclui um sistema operacional e uma estrutura para aplicações gerais e aplicações já existentes no seu ambiente. As aplicações gerais são escritas na linguagem Java através da API fornecida pela Google, com o *Software Development Kit* (SDK) (LEAVIT, 2011), (SHASBTAI, et al 2010).

Nas camadas de software do Android (observada de baixo para cima na Figura 1.), tem-se:

a.) A camada básica que é o Kernel Linux (versão 2.6) que aciona os *drivers* do dispositivo, gerenciamento de memória, gerenciamento de processos e rede (*Networking*) (LEAVIT, 2011), (SHASBTAI, et al 2010).

b.) Na camada seguinte, que inclui a *System Native Libraries* e a JVM DALVINK – *Android Runtime*, tem-se as bibliotecas nativas do ambiente escritas em C/C++, passíveis de serem utilizadas e/ou estendidas, através do *Native Development Kit* (NDK) também fornecido pela Google.

Estas bibliotecas são utilizadas por vários componentes do sistema nas camadas de software. Incorporadas através de interfaces Java, elas oferecem uma biblioteca de componentes C personalizados, um mecanismo de banco de dados SQL, uma biblioteca gráfica 2D e 3D, e mecanismos de *browser web* (WebKit) e *codecs* de mídia, como exemplo, MPEG-4 e MP3 (LEAVIT, 2011), (SHASBTAI, et al 2010).

Ainda nesta mesma camada, tem-se o Android Runtime, que consiste na máquina virtual para as aplicações Java a Dalvink (que é uma JVM otimizada para execução em dispositivos móveis) e as bibliotecas do núcleo da API Java para o desenvolvimento das aplicações.

No desenvolvimento de aplicações para Android, utilizam-se linguagem Java e normalmente todos os seus recursos, mas depois que o bytecode (.class) é compilado, ele é convertido para o formato .dex (*Dalvink Executable*), com arquivos que são mais compactos e eficientes do que arquivos de classe Java convencional *(fato relevante para dispositivos alimentados por bateria e memória limitada),* que representam a aplicação compilada do Android.

Os arquivos .dex e outros recursos como imagens, são compactados em um único arquivo com a extensão .apk (*Android Package File*), que representa a aplicação final, pronta para ser distribuída e instalada (LEAVIT, 2011), (SHASBTAI, et al 2010).

Observa-se ainda na arquitetura, que quando se utiliza uma IDE (*Integrated Development Environment*), como a plataforma *open source* Eclipse, por exemplo, toda essa compilação e geração do arquivo .apk, ocorre automaticamente.

A camada *Android ApplicationFramework* que é composta pela API Java, é uma biblioteca escrita em Java e fornece um subconjunto significativo de pacotes Java SE 5, como por exemplo: as coleções padrão, I/O, rede, utilitários e algumas bibliotecas especificas do Android. Estas bibliotecas são necessárias para acessar os recursos do hardware, o sistema operacional e o que as bibliotecas nativas oferecem nesta camada (escrita totalmente em Java). Nas bibliotecas estão inclusas ferramentas fornecidas pela Google e extensões proprietárias ou serviços.

Um componente importante deste quadro é o gerenciador de atividade, que atesta e verifica o ciclo de vida das aplicações (LEAVIT, 2011), (SHASBTAI, et al 2010).

Na última Camada das Aplicações (*Applications*) cada aplicação é empacotada num arquivo .apk (que é semelhante ao arquivo .jar, do Java padrão), detendo todo código fonte e outros recursos, como por exemplo, imagens e o arquivo manifesto.xml.

Esta é a camada de implementação de aplicações gerais, tais como: um telefone, *web browser*, cliente de e-mail e todas as outras. (LEAVIT, 2011), (SHASBTAI, et al 2010).

Cada aplicação .apk está associada a um único processo do sistema operacional, no qual todos os componentes desta aplicação (atividades, serviços, receptores de radiodifusão e fornecedores de conteúdo) são executados junto com recursos, permissões e requisitos, devendo ser listados no arquivo AndroidManifest.xml, para que ocorram as permissões de utilização dos recursos do Android (LEAVIT, 2011), (SHASBTAI, et al 2010).

Assim, o Android é uma coleção de componentes e um sistema multiprocessado, onde cada aplicação .apk e partes do sistema é executada em seu próprio processo, além do fato de que uma aplicação .apk só poderá se comunicar com outra aplicação .apk através de mecanismos fornecidos pelo próprio sistema operacional, já que componentes de uma aplicação .apk são isolados dos componentes de outra aplicação .apk (LEAVIT, 2011), (SHASBTAI, et al 2010).

2.2 Mecanismo Padrão de Proteção do Sistema Operacional Android

No mecanismo padrão de proteção, verifica-se que em grande parte, o controle de proteção entre as aplicações e o sistema operacional é executado no nível do processo, através da instalação padrão do Linux, com o usuário e grupos de identificadores (IDs) atribuídos para estas aplicações (LEAVIT, 2011), (SHASBTAI, et al 2010).

O controle de acesso das aplicações é fornecido através de um mecanismo de permissão, que impõe restrições sobre operações específicas, que uma determinada aplicação pode ou não executar nesta plataforma (LEAVIT, 2011), (SHASBTAI, et al 2010).

O acesso para os arquivos (relação de aplicações e arquivos) no Android, é proporcionado pelo mecanismo padrão de permissão do Linux. Para cada pacote instalado e usuários de arquivos de Android .apk (Aplicação) em um dispositivo móvel é dada sua própria identificação Linux ID de usuário POSIX (*Portable Operationg System Interface*) (LEAVIT, 2011), (SHASBTAI, et al 2010).

Cada arquivo, está associado ao proprietário pelo identificador de usuário (ID), IDs de grupo e três tuplas de permissão: leitura, escrita e execução (rwx). Esta estrutura de proteção, fornece acesso a arquivos, diretórios, drivers, terminais, sensores de hardware, mudanças de estado de alimentação, de áudio, entrada direta de leituras, memória compartilhada e acesso também aos componentes e processos internos (conhecidos como Daemon em Linux) (LEAVIT, 2011), (SHASBTAI, et al 2010).

Dentro do Kernel Linux do Android, esse ID é atribuído quando a aplicação é instalada no dispositivo e, como conseqüência, o código de dois pacotes diferentes não podem ser executados no mesmo processo do sistema operacional, impedindo assim, que uma aplicação interfira em outra aplicação e vice-versa (LEAVIT, 2011), (SHASBTAI, et al 2010).

Para que duas aplicações possam compartilhar o mesmo conjunto de permissões em um processo do Android, elas devem compartilhar o mesmo ID. Esta ação só é permitida através da característica sharedUserID, sendo que as duas aplicações deverão declarar seu uso e possuir a mesma assinatura digital (LEAVIT, 2011), (SHASBTAI, et al 2010).

Pode-se supor o aparecimento de um invasor com habilidade de escrever arquivos em qualquer lugar do sistema operacional. Ele teria negada a possibilidade de substituir arquivos críticos, pois a imagem do Linux é montada somente para leitura e, para poder remontar a imagem, ele precisaria ter acesso como usuário "*root*" que é a permissão de acesso exclusiva do proprietário da instalação (LEAVIT, 2011), (SHASBTAI, et al 2010).

No mecanismo de proteção padrão do Android tem-se a chamada proteção de TIPO, que é uma propriedade das linguagens de programação e força o conteúdo de variável a aderir a um formato específico, impedindo o uso errado ou indesejável (LEAVIT, 2011), (SHASBTAI, et al 2010).

A falta de tipo e/ou a verificação de limites, podem levar a corrupção de memória e ataques de estouro de *buffer*, que são um dos meios para a execução de código arbitrário ou malicioso (LEAVIT, 2011), (SHASBTAI, et al 2010).

A linguagem Java, que é a linguagem empregada no Android, é fortemente tipada os programas escritos somente em Java, são menos suscetíveis a execução de código arbitrário ou malicioso. Porém, o Android permite também a utilização da linguagem C como código nativo, liberando assim a ligação de código (partes do código Java das aplicações escrito em C ou C++), sem a verificação de tipo e, caso não seja especificamente descrito pelo programador no código C/C++, não existirão fronteiras de verificação, expandindo potencialmente o risco de proteção a camada padrão de proteção. Na máquina virtual Java Dalvik VM, cada aplicação é executada em sua própria máquina virtual e isto impede a ocorrência de *overflow* de *buffer* por código, execuções remotas e de pilha de forma arbitrária ou maliciosa (LEAVIT, 2011), (SHASBTAI, et al 2010).

Muitas bibliotecas nativas escritas em C/C++ (Linguagem sem tipos fortes passíveis de problemas de proteção), são utilizadas pelo Android. Estas bibliotecas são utilizadas por processos nativos, por outras bibliotecas nativas ou pela máquina virtual Dalvink através de *Java Native Interface* (JNI) (LEAVIT, 2011), (SHASBTAI, et al 2010).

JNI é um método para chamar métodos nativos no código Java (Em C/C++ por exemplo), no contexto do mesmo processo. As bibliotecas nativas são alvos de busca de vulnerabilidade de proteção pelos invasores maliciosos (hacker) (LEAVIT, 2011), (SHASBTAI, et al 2010).

O coração da proteção padrão das aplicações no Android é o sistema de

permissões, que impõe restrições a operações específicas que um aplicativo pode executar. O componente de gerenciamento da plataforma é responsável pela concessão de permissões para aplicações na instalação da .apk e a camada de Android Application Framework é responsável por fazer cumprir as permissões do sistema operacional em tempo de execução detectando aplicações maliciosas (LEAVIT, 2011), (SHASBTAI, et al 2010).

Existem cerca de 100 permissões *built-in* (Permissões Padrão) em operações que controlam o Android, incluindo a marcação do telefone (CALL_PHONE), captação de imagens (Câmera), acesso a Internet (INTERNET), cursos em áudio (READ_INPUT_STATE) ou escrita de um SMS (WRITE_SMS). Estas permissões possuem níveis de proteção associados e qualquer aplicação pode declarar permissão adicional (LEAVIT, 2011), (SHASBTAI, et al 2010).

Para obter permissões para seu funcionamento, uma aplicação deve solicitar explicitamente sua intenção em seu arquivo manifesto (manifesto.xml – o "contrato" entre o sistema operacional e o aplicativo Android) e estão classificadas como:

- ✓ Normal: permissões que não são especialmente perigosas;
- ✓ *Dangerous*: permissões que são mais perigosas do que o normal;
- ✓ Assinatura: permissões que só podem ser concedidas a outros pacotes que são assinados com a mesma assinatura, como a declarada na permissão;
- ✓ SignatureOrSystem: permissão de assinatura que também é concedida aos pacotes instalados na imagem do Android (LEAVIT, 2011), (SHASBTAI, et al 2010).

Na instalação de uma aplicação, as permissões solicitadas pelo aplicativo são concedidas a ele com base na verificação das assinaturas declaradas na aplicação, e na interação com o usuário que autoriza as permissões (LEAVIT, 2011), (SHASBTAI, et al 2010).

Após a instalação da aplicação suas permissões são fixadas e não poderão sofrer mais modificações (LEAVIT, 2011), (SHASBTAI, et al 2010).

Como descrito em Leavit, (2011) e Shasbtai, (2010), podem-se reunir em três grupos o mecanismo de proteção padrão do Android: Mecanismos de proteção Linux, Recursos e Mecanismos de proteção específicos do Sistema Operacional Android (como especificado na Tabela 1).

Tabela 1. - Grupos de mecanismos de proteção padrão do Android.

Descrição dos Mecanismos de Proteção		
1) Mecanismos Linux	**2) Características do Ambiente (Recursos)**	**3) Mecanismos específicos da plataforma**
Usuários POSIX Cada aplicação .apk está associada a uma diferente UID. Isto impede um aplicativo de interferir em outro. O acesso ao arquivo e diretório da aplicação está disponível apenas para o "proprietário" da aplicação e impede que um arquivo de uma aplicação acesse outro de outra aplicação.	**Unidade de gestão de memória:** Cada processo esta sendo executado em seu próprio espaço de endereço, a escalação de privilégios, a divulgação de informação, e ataques DoS são evitados. Segurança de tipo é onde a aplicação de conteúdo variável para aderir a um formato específico, tanto na compilação e em tempo de execução é controlada, impede *overflows de buffer* e quebra de pilha; **Recursos de segurança da Operadora de Telefonia Móvel:** Utilizando o cartão SIM para autenticar e autorizar a identidade do usuário e roubo de telefone.	**Permissões da aplicação:** Cada aplicação declara qual é a permissão exigida para seu funcionamento no momento da instalação (AndroidManifest.xml), as habilidades da aplicação são limitadas para evitar comportamento malicioso; **Componente de Encapsulamento:** Cada componente em uma aplicação (por exemplo, Atividade ou serviço) tem um nível de visibilidade que regula o acesso de outras aplicações a ela (ex: ligação a um serviço), impede que um aplicativo interfira em outro acessando componentes privados ou APIs; **Assinatura da aplicação:** Arquivos de aplicações .apk são assinados pelo desenvolvedor e verificado pelo gerenciador de pacotes e correspondência. Verifica que duas aplicações são da mesma fonte. **Maquina Virtual Dalvink:** Cada aplicação é executada em sua própria máquina virtual. Isto impede código de buffer overflows, execução remota e de pilha.

O Kernel Linux não é bem visto por sua estrutura de proteção. Em 2007, 83 CVE (Common Vulnerabilities and Exposures) foram registradas e em 2008, 73 CVE.

Uma vulnerabilidade no kernel do Linux é explorada pelo atacante malicioso (hacker) para obter a senha de acesso de administrador da máquina "acesso

root". Mecanismos de proteção adicional à camada padrão devem ser acrescentados para garantir ou minimizar este tipo de vulnerabilidade (LEAVIT, 2011), (SHASBTAI, et al 2010).

A Google publicou o código fonte das bibliotecas nativas do Android que provoca maior vulnerabilidade de segurança e em sua estrutura de proteção, pois todos os defeitos são conhecidos neste sistema operacional. Outros exemplos de ataques ao Android, podem ser verificados em detalhes em (LEAVIT, 2011), (SHASBTAI, et al 2010).

Existem ainda mecanismos incorporados a proteção padrão que já são utilizados por operadores de serviços para mobilidade, como é o caso, por exemplo, do cartão SIM de telefonia, que contém um segredo compartilhado apenas pelo cartão e a operadora, um recurso clássico de proteção de telefonia móvel (LEAVIT, 2011), (SHASBTAI, et al 2010).

Faz-se também necessário salientar, que o hardware pode ser atacado, pois os dispositivos móveis tem vários componentes vulneráveis. O Android possui a tendência de permitir que as aplicações tenham acesso ao máximo possível dos recursos do sistema operacional, o que possibilita uma maior vulnerabilidade também para ataques ao hardware (LEAVIT, 2011), (SHASBTAI, et al 2010).

Cartões de armazenamento (Storage Device SD), por exemplo, podem ter sua gravação magnética desgastada uma vez que têm um número finito de ciclos de apagar e o Android não limita a taxa de Entrada/Saída ou define uma quota limite de Entrada e Saída (LEAVIT, 2011), (SHASBTAI, et al 2010).

A drenagem de energia da bateria do dispositivo pode ser provocada pelo invasor ao manter o processador (CPU) rodando ou mantendo uma vigília presa (*lock*) e não é fácil adicionar proteção em tais casos (LEAVIT, 2011), (SHASBTAI, et al 2010).

Hoje, a drenagem de bateria é objeto de vários artigos e pesquisas para garantia da proteção nestas situações (LEAVIT, 2011), (SHASBTAI, et al 2010).

A drenagem da energia da bateria do dispositivo é um dos elementos importantes deste trabalho e será melhor descrita no capitulo 3.

Na avaliação dos riscos identificados por Shasbtai, et al (2010) e que devem ser combatidos com adição de proteção complementar no sistema operacional padrão Android, foram definidos os cinco grupos mais importantes de ameaças que são descritos a seguir:

- Grupo de ameaças do Tipo 1: Definido como a ação de comprometer a confidencialidade, disponibilidade e/ou integridade por atividade maliciosa, usando as permissões concedidas a um aplicativo instalado. Este cenário de ataque é provável que aconteça potencialmente e tem um alto impacto no dispositivo (LEAVIT, 2011), (SHASBTAI, et al 2010);

- Grupo de ameaças do Tipo 2: Definido como a ação de comprometer a confidencialidade, disponibilidade e/ou a integridade de uma aplicação explorando uma vulnerabilidade no kernel do Linux ou bibliotecas do sistema. Este cenário foi provado possível e a análise mostra que a proteção contra vulnerabilidades adicionais é susceptível de ser encontradas. Embora, ela tenha uma baixa probabilidade de ocorrência, ela carrega um potencial para causar danos graves (LEAVIT, 2011), (SHASBTAI, et al 2010);

- Grupo de ameaças do Tipo 3: Definido como a ação de comprometer a disponibilidade, confidencialidade e/ou integridade de conteúdo privado/confidencial. O conteúdo do cartão SD, por exemplo, não esta protegido por qualquer mecanismo de controle de acesso. Além disso, a comunicação sem fio pode ter o mecanismo de controle de acesso remotamente acessado (LEAVIT, 2011), (SHASBTAI, et al 2010);

- Grupo de ameaças do Tipo 4: Definido como as ações com recursos de drenagem de energia da bateria do dispositivo móvel. Ações que utilizam a propriedade de não existir espaço em disco e nem quota de memória (RAM) por Aplicação. São possíveis monopolizando o processador

(CPU) através de várias técnicas (LEAVIT, 2011), (SHASBTAI, et al 2010);

- Grupo de ameaças do Tipo 5: Definido como o comprometimento de uma rede interna/protegida. Os dispositivos móveis com Android podem ser usados para atacar outros dispositivos, computadores ou redes por execução de scanners de rede ou porta, SMS/ MMS/*worms* de e-mail e vários outros métodos de ataque (LEAVIT, 2011), (SHASBTAI, et al 2010);

As propostas para adição de proteção ao mecanismo de proteção padrão do Android, para os cinco grupos de ameaças citados acima, foram avaliados por Shasbtai, et al (2010) no nível de mitigação e contramedidas para cada grupo como descrito abaixo:

- Solução proposta para o Grupo de ameaça do Tipo 1: Definido como comprometer a disponibilidade e/ou de confidencialidade e/ou integridade maliciosa por utilização das permissões concedidas a uma aplicação instalada.

  Sistema de Detecção de Intrusão (*Intrusion Detection/Prevention System*) é uma solução IDS, é bem adequada para definir o comportamento normal do sistema, da aplicação ou do usuário e detectar desvios, e/ou alternativamente, detectar padrões de comportamento malicioso de malware. IDS também pode servir como ferramenta eficaz na descoberta de ameaças inicialmente desconhecidas e isoladas. No entanto, uma vez que o *malware* pode adaptar-se rapidamente e mascarar seu comportamento de acordo com as ferramentas de proteção capazes de detectar isso, a eficácia do IDS pode diminuir ao longo do tempo.

  Firewall - O *firewall* é uma solução para a rede de ataques relacionados. Pode impedir vazamento de dados por um *malware* instalado. No entanto, nem todos os ataques são baseados nas permissões de abuso de acesso em rede e, portanto, um *firewall* seria muito útil contra um

conjunto parcial de ataques.

Aplicação de Certificação - A certificação é uma contramedida eficaz contra aplicações maliciosas. Como cada aplicação teria que ser exaustivamente testada e revisada antes da certificação, a autorização para utilizar qualquer das funções do dispositivo, aplicações maliciosas deveriam ser apanhadas na sua fase inicial e não poderiam receber uma certificação adequada. Infelizmente, com a certificação de aplicação, os custos incorridos pela necessidade de estabelecer e manter o provedor de certificado, modificar a sistema operacional Android existente para suportar as funcionalidades requeridas e verificar cada aplicação são bastante elevados. Permissões seletivas no Android proporcionam a capacidade de aprovar apenas um subconjunto de permissões para um aplicativo instalado reduziria o risco de forma maliciosa usando permissões concedidas. Esta solução é mais adequada para usuários avançados, pois os usuários ingênuos ainda podem instalar aplicações sem validar as permissões solicitadas.

Análise automatizada estática e Código de Verificação: fornecer a capacidade de avaliar automaticamente a natureza de uma aplicação, suas capacidades e a diferença entre o que o aplicativo pode fazer e como fazer reivindicações (LEAVIT, 2011), (SHASBTAI, et al 2010);

- Solução proposta para o Grupo de ameaça do Tipo 2: comprometer a disponibilidade e/ou de confidencialidade e/ou a integridade de uma aplicação explorando uma vulnerabilidade no kernel do Linux ou bibliotecas do sistema.

  Controle de Acesso do SOLinux é bem adequada para limitar a capacidade das entidades no sistema operacional. O potencial de risco de vulnerabilidades exploráveis poderia levar à uma situação em que o todo poderia ser prejudicado se os invasores forem capazes de obter privilégios de super-usuário. Ao limitar a capacidade dos processos de raiz e entidades de outra forma potencialmente vulneráveis ou de alta prioridade, o SOLinux impediria o invasor de forçar o sistema a fazer o

seu/sua licitação e, assim, tornar o ataque muito menos eficaz. No entanto, uma vez que cada entidade requer a capacidade de executar certos comandos para o seu funcionamento normal, esses comandos não devem ser bloqueadas pelo SOLinux. Caso as capacidades de execução da entidade não tenham sido comprometidas, o invasor ainda teria o mesmo espaço de manobra para desencadear um ataque. Em outras palavras, o ataque só poderia ser parcialmente mitigado (LEAVIT, 2011), (SHASBTAI, et al 2010);

- Solução proposta para o Grupo de ameaça do Tipo 3: Comprometimento disponibilidade e/ou de confidencialidade e/ou integridade de conteúdo privado/confidencial.

  Acesso - A exigência de fornecer uma senha, a fim de desbloquear certas funções do dispositivo é uma ferramenta bem conhecida e eficaz contra uma variedade de ameaças, em particular, a exposição de conteúdo privado. No caso, se o dispositivo é roubado com o bloqueio no local, um invasor não seria capaz de acessar qualquer uma das informações privadas sem a senha. No entanto, se o dispositivo é roubado após ser desbloqueado, o mecanismo de defesa torna-se inútil e daria ao invasor o poder de fazer o que ele/ela desejasse com o dispositivo.

  Firewall - Um *firewall* pode muito bem proteger contra vazamentos de informações através de qualquer interface de Rede. Usando um apátrida (Individuo que não é titular de nenhuma nacionalidade) ou a inspeção *stateful* (inspeção do *Firewall* que mantém seu estado) de conteúdo no meio de comunicação, poderia decidir se a informação confidencial está sendo enviada e bloquear a comunicação. Como o *firewall* opera em níveis mais baixos do *kernel*, ele não poderia ser ignorado por aplicações mal-intencionados (na ausência de vulnerabilidades exploráveis no *kernel* do Linux ou bibliotecas do sistema). Ele também pode trabalhar em conjunto com um mecanismo de controle de acesso, tais como o SOLinux, para proporcionar um nível de proteção ainda

maior. No entanto, as interfaces de rede não são o único caminho que o malware que pode tomar a fim de realizar um vazamento de dados privados a partir do dispositivo. Uma abordagem alternativa seria para enviar os dados através de mensagens SMS/MMS. Infelizmente, o *ontextfirewall* não poderia bloquear tal abordagem.

Criptografia de Dados - A criptografia de dados é um excelente meio para combater a exposição de dados privados. Como só o proprietário conhece a chave que é capaz de decifrar tais dados, a informação seria protegida em caso de exposição. Mesmo que o aparelho fosse roubado e que o invasor tivesse acesso total a todas as informações, ele não seria capaz de decifrar a criptografia num tempo razoável.

Controle de *Acesso Context Awa*re - Um mecanismo de controle de acesso sensível ao contexto (*Context-Aware Access Control*) poderia limitar o acesso a dados privados, dependendo do contexto em que o dispositivo se encontra com base em sua localização, rede de celular, se estiver conectado à Internet *Wi-Fi* e outros mais . Esse mecanismo pode se defender contra uma variedade de ataques, dependendo de divulgação de informações. No entanto, sob estas circunstâncias, se o ataque ocorrer enquanto o dispositivo estiver em um contexto que permite o acesso à informação, o acesso será permitido e as informações divulgadas. Num outro exemplo, se o dispositivo é roubado e transferido para um local externo, os dados estariam seguros e inacessíveis para o invasor.

Gerenciamento Remoto - Capacidades de gerenciamento remoto são severamente limitadas. No entanto, quando combinada com soluções de proteção adicionais, como um *firewall* ou sensíveis ao contexto de proteção, o potencial de proteção aumenta substancialmente. Se o dispositivo é roubado, as informações poderiam ser protegidos por se ligar remotamente a um mecanismo de defesa. Mesmo durante a operação diária do dispositivo, se o gerente remoto é capaz de identificar um verme (*worm*) rondando a rede celular / sem fio, ele pode configurar o *firewall* de acordo, para bloquear o verme, a fim de evitar

qualquer divulgação de informações. Todos os itens acima dependem de intervenção humana durante o curso de um ataque ou anterior a ele. Além disso, para defender contra ataques no momento certo, um meio constante de monitoramento do dispositivo faz-se necessário. Tal exigência é susceptível e pode ser dispendiosa em termos de recursos do dispositivo e ainda exigente para o gerenciamento remoto (LEAVIT, 2011), (SHASBTAI, et al 2010).

- Solução proposta para o Grupo de ameaça do Tipo 4: Drenagem dos recursos

  Gerenciamento de Recursos - O recurso de solução de gerenciamento de proteção reduz a ameaça de drenagem de recursos por aplicações mal-intencionadas. O funcionamento do mecanismo consiste de boa alocação de recursos para aplicações de acordo com suas necessidades e tem em conta sua importância, por exemplo, aplicação de telefone é muito importante e, portanto, deve receber mais processador (CPU) do que um jogo. Nesse caso, a drenagem de recursos sem supervisão não é possível. Se os espaços definidos de armazenamento em disco são mantidos e a taxa de rede é limitada e, ainda for permitido até um determinado espaço de uso em disco, um ataque de negação de serviço pode ser mitigado através desta supervisão. Porém, devido à dificuldade de implementação dessa solução de proteção, sua aplicabilidade é bastante baixa.

  Sistema de Detecção de Intrusão / Prevenção - UM *host-based* IDS pode contrariar a drenagem maliciosa de memória da bateria ou processador (CPU) através da detecção de alterações na taxa de anormal nos níveis de recursos. Na prática, qualquer *malware* pretende permanecer sem ser detectado e com isso, devemos ter o perfil de uso normal do usuário continuamente mantido e validado (LEAVIT, 2011), (SHASBTAI, et al 2010).

- Solução proposta para o Grupo de ameaça do Tipo 5: Comprometer rede interna / protected

  Rede Privada Virtual - A rede privada virtual (*Virtual Privite Network* - VPN). Esta solução baseia-se em princípios maduros como códigos de mensagem de autenticação e criptografia para proteger a comunicação.

  Gerenciamento Remoto - A execução de uma política de proteção ao lidar com redes internas protegidas pode muito facilmente ser feita por uma estrutura de gestão centralizada e remota, que é controlada pelo administrador da rede. No entanto, a eficácia da mitigação de ameaças depende da vigilância do administrador, ou seja, um fator humano que é uma brecha na segurança da solução.

  Controle de Acesso *Context Aware* - Ao lidar com redes protegidas, o controle de acesso sensível ao contexto pode de fato ser visto como uma versão automatizada da abordagem de gerenciamento remoto. Sobre o detecção de um contexto que envolve uma conexão ativa com a rede protegida, o mecanismo de Controle de Acesso *Context Aware* pode aumentar as medidas de proteção ativa no dispositivo. Tais medidas podem incluir a criptografia de conexão, autenticação e muito mais (LEAVIT, 2011), (SHASBTAI, et al 2010).

Baseado nos argumentos apresentados em Leavit, (2011) e Shasbtai, et al (2010), a camada padrão de proteção do Android vem se apresentando insuficiente e insatisfatória, pois tem deixando o sistema cada vez mais vulnerável à ataques a sua proteção.

Uma busca por recursos técnicos de aperfeiçoamento para complementar a camada padrão de proteção deste sistema, apresenta-se como necessária não só para a manutenção da proteção dos dispositivos móveis que o utilizam e suas aplicações, mas também, de sua liderança no mercado (LEAVIT, 2011), (SHASBTAI, et al 2010).

2.3 Técnicas de Proteção e Controle de Utilização com Ativação de Regras Sensíveis ao Contexto

O objetivo do controle de acesso sensível ao contexto é melhorar a eficácia e proporcionar um maior grau de refinamento para a camada padrão de proteção do Android, possibilitando assim, uma melhor experiência ao usuário através da inclusão desta proteção adicional, levando em consideração os seguintes tópicos: tipos de ataque possíveis, minimização na perda de desempenho e maximização dos recursos, como exemplo, consumo da bateria, nos mais diferentes níveis de proteção.

A técnica conhecida como *Context-Aware Computing* ou computação sensível ao contexto no Android permite atingir esse objetivo.

A técnica de *Context-Aware* permite uma melhor proteção de conteúdo confidencial e garantia de integridade para vários serviços, uma vez que a maioria dos trabalhos de pesquisa para impor políticas de proteção em dispositivos móveis, apresenta políticas que são consideradas mal elaboradas, com pouca flexibilidade e controle do dispositivo pelo usuário final (por exemplo: onde o usuário pode somente permitir que um aplicativo possa ser executado ou não) (RODRIGUEZ, CROWCROFT, 2011), (LEAVIT, 2011), (SHASBTAI, et al 2010), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Essa proteção deve ser combinada com um gerenciamento eficiente de energia da bateria para prover melhor experiência ao usuário (RODRIGUEZ, CROWCROFT, 2011), (LEAVIT, 2011), (SHASBTAI, et al 2010), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

2.4 Mecanismos para Adicionar ao Android Gerenciamento de Energia

A utilização de técnicas de controle do uso de dispositivos móveis para proteção por regras sensíveis ao contexto, envolve uma complexidade maior na utilização dos recursos do dispositivo móvel, em particular o processamento e o consumo de energia da bateria. Como resultado, é bastante possível para uma aplicação que consome muita energia a possibilidade de reduzir

drasticamente o tempo de operação do dispositivo (RODRIGUEZ, CROWCROFT, 2011), (CONTI, NGUYEN, CRISPO, 2010).

Com a integração de múltiplos componentes de hardware e softwares, disponíveis para dispositivos móveis, é possível melhorar sua funcionalidade e experiência do usuário, porém com prejuízo e redução da vida útil da bateria em poucas horas. Neste aspecto, apesar do esforço para otimizar o Android, um gerenciamento de energia eficiente é fundamental, pois modelos de negócio complexos, como uma camada adicional de proteção pelo controle de uso por regras sensíveis ao contexto, por exemplo, em plataforma móveis, estão ficando cada vez mais comuns. Fato este, que compromete o gerenciamento padrão do consumo de energia. Assim sendo, há grande espaço para melhorias no Android para este tipo de gerenciamento. (RODRIGUEZ, CROWCROFT, 2011).

Estudos como o de Rodriguez e Crowcroft (2011) propõem adicionar ao sistema operacional base, um sistema operacional adicional para gerenciamento de energia. Com relação a este fato, esta camada adicional seria centrada na utilização do dispositivo pelo usuário e ao seu consumo de bateria, através do gerenciamento dos recursos de forma próativa e por exploração de acesso oportunista aos recursos em dispositivos próximos. Utilizando conexões sociais entre os usuários, esta proposta é chamada de Erdos.

As atuais plataformas móveis integram sensores como GPS, vários tipos de interfaces sem fio (em Giga Hertz), processadores (CPU) multinúcleos e ecrã táctil (Telas de toque). Esta tendência provocou o aparecimento de aplicações móveis ricas e que apesar de melhorar a experiência do usuário na utilização do dispositivo, podem vir a tornarem-se sumidouros de energia, dependendo do modo como estes usuários interagem com seus aparelhos (RODRIGUEZ, CROWCROFT, 2011).

O estado da arte de bateria de lítio-íon, indica claramente que sua capacidade ainda será limitada por parâmetros de projeto, tais como: o tamanho e peso da bateria, de acordo com Rodriguez e Crowcroft (2011). Os autores acreditam

que uma das razões por trás da pouca eficiência do controle do uso de energia em dispositivos móveis, é que os sistemas operacionais atuais não controlam naturalmente o acesso a energia que é consumida pelos recursos utilizados por suas aplicações (RODRIGUEZ, CROWCROFT, 2011).

Portanto, o sistema operacional Erdos é uma extensão do Android e o gerenciamento de sua energia funciona de forma transparente e, suas aplicações são capazes de acessar recursos normalmente (RODRIGUEZ, CROWCROFT, 2011).

As demandas de energia em dispositivos móveis são causadas por componentes de hardware, que podem operar simultaneamente e responder à aplicações em execuções.

Ações de usuários geradas por ações sociais (ex: um envio de *e-mail* ou telefonema) e hábitos dos usuários (carregamento da bateria do aparelho em casa). Estas possibilidades permitem "mapear" os recursos das aplicações com base na demanda das atividades dos usuários, que podem ser inferidas a partir de informações sensíveis ao contexto de utilização do dispositivo móvel (RODRIGUEZ, CROWCROFT, 2011).

O sistema operacional Erdos aprende através do comportamento do usuário e suas atividades com base em um algoritmo de tempo de utilização e a informação contextual reunida com relação a energia passiva (energia original da bateria do dispositivo versus a energia restante) disponível anteriormente no dispositivo (RODRIGUEZ, CROWCROFT, 2011).

Não necessariamente, este sistema operacional de gerenciamento de energia, precisa saber o significado real das atividades, mas deverá ser capaz de diferenciar situações estacionárias em locais definidos e tempos e, também, as transições entre eles (ex: identificar quando o usuário está no seu local de trabalho, em sua casa ou viajando entre esses lugares, além de contar quais são os recursos destes estados do dispositivo e qual a exigência para esses cenários) (RODRIGUEZ, CROWCROFT, 2011).

No caso do Erdos, estará disponível um monitor de recursos sensíveis ao

contexto que verificará a utilização dos recursos do dispositivo. As aplicações executadas, os recursos de demandas de energia destas aplicações, podem ser diferentes em cada uma dessas atividades, bem como, alterações no estado dos recursos (quando o dispositivo está conectado em uma fonte de energia). Estas ações são altamente previsíveis (RODRIGUEZ, CROWCROFT, 2011).

O Erdos pode prever situações e os recursos demandados dos usuários, e deverá ser capaz de lidar com a incerteza, a fim de identificar novas atividades e mudanças nos padrões dos usuários (quando uma aplicação nova está instalada no dispositivo ou quando o usuário está num novo local) (RODRIGUEZ, CROWCROFT, 2011).

De fato, a alocação de recursos para aplicações de forma pró-ativa com base em informações contextuais, associadas aos hábitos dos usuários podem economizar considerável quantidade de energia, sendo permitido que o usuário decida, se quer ou não ativar os recursos automáticos para gerenciamento de energia (RODRIGUEZ, CROWCROFT, 2011).

Usuários podem preferir obter *feedback* do sistema sobre futuras limitações ou indisponibilidade de energia e assim, obter recursos para se adaptar a maneira como eles interagem com os seus aparelhos, ao invés de permitir recursos automáticos de gerenciamento, com objetivo de prolongar a vida útil da bateria (RODRIGUEZ, CROWCROFT, 2011).

Existem situações cotidianas, nas quais as aplicações móveis podem drenar a bateria devido ao inadequado consumo dos recursos. Uma atividade comum, como a sincronização do cliente de *e-mail*, é um bom exemplo. Realizar este tipo de ação, pode ser de alto dispêndio energético, uma vez que acionar o processador (CPU) e a interface de conectividade, quando o aparelho está no modo inativo, seguido por um pedido de DNS (*Domain Name System*) e uma conexão com o servidor de *e-mail*. Tal ação acontece regularmente, mesmo em situações quando a sua execução não melhorar a experiência do usuário na utilização do sistema (ex.: à noite, quando o usuário está dormindo) e em cenários onde pode-se muito bem saber que não há cobertura de rede nas

experiências anteriores (RODRIGUEZ, CROWCROFT, 2011), (CONTI, NGUYEN, CRISPO, 2010).

Por outro lado, as experiências anteriores sobre a mobilidade humana e interação social por meio de varreduras *Bluetooth*, indicam que há muitas oportunidades para estabelecer conexões oportunistas entre os dispositivos.

Um dispositivo móvel, pode claramente economizar energia, mesmo acessando um recurso como o GPS a partir de um telefone próximo, ao invés de acessar o receptor local GPS (embora o dispositivo de compartilhamento do recurso sacrifique-se a curto prazo, a fim de compartilhar seus recursos para outros). É possível alcançar benefícios em termos temporais, executando uma varredura *Bluetooth* e conectando-se com um dispositivo próximo, o que levaria 11,5 segundos em média, ao recuperar a posição sobre o receptor GPS, que pode levar 4 segundos ou até minutos, dependendo da disponibilidade dos dados orbitais dos satélites GPS (RODRIGUEZ, CROWCROFT, 2011).

Partindo da perspectiva de técnicas de controle de utilização do dispositivo móvel, sensível no contexto e controle de uso, para fins de proteção e melhora da experiência do usuário, verifica-se que a solução de um sistema operacional de gerenciamento de energia (por exemplo, o Erdos), integrada ao Android, torna possível um maior número e complexidade de regras sensíveis ao contexto ou conjunto de regras (políticas de proteção em dispositivos móveis) para proteção de controle de uso do dispositivo móvel (RODRIGUEZ, CROWCROFT, 2011), (CONTI, NGUYEN, CRISPO, 2010).

2.5 Definição de uma Melhor Experiência do Usuário

Em todo o trabalho é usada a expressão "obter uma melhor experiência do usuário". Em termos de proteção e controle de utilização do dispositivo móvel no caso, do Android, entende-se que o usuário depende de suas decisões ao autorizar a instalação de cada aplicação. O acesso que esta aplicação fará aos recursos do dispositivo é autorizado pelo usuário. Muitas aplicações disponíveis para serem baixadas não possuem garantias de proteção para o dispositivo contra códigos maliciosos. O controle de utilização de recursos do

dispositivo usualmente também não é acessível ao usuário. Conforme os trabalhos estudados, esta situação não proporciona uma adequada experiência do usuário na utilização do seu dispositivo móvel.

Para o presente trabalho considera-se que para a obtenção de uma adequada experiência do usuário de forma protegida e com controle de utilização do dispositivo móvel, incluindo-se aqui, o nível adequado de consumo de energia da bateria, é quando o dispositivo esta protegido, controlado e disponível para atender as necessidades objetivas deste usuário.

Por necessidades objetivas do usuário, entende-se que as regras que num determinado contexto devem ser observadas deverão proporcionar o resultado desejado. Em termos de disponibilidade de utilização (tempo disponível de energia da bateia) e uma proteção desejada em particular, a mesma sendo baseada na localização geográfica e no contexto adequado para esta posição.

2.6 Considerações do Capítulo

Apresentamos o conceito e descrição da utilização da técnica de proteção e controle do uso de dispositivos móveis com o sistema operacional Android baseada na ativação de regras sensíveis ao contexto.

A utilização da técnica acima, tem como finalidade a melhoria da experiência do usuário no sentido de proteção e controle de utilização do dispositivo de forma adequada e com disponibilidade em termos de energia da bateria suficiente e maximizada desta experiência.

No próximo capítulo será descrito o problema e a necessidade da adição de proteção e controle de uso no Android contextualizando as técnicas acima para a solução deste problema.

3 ANÁLISE DO PROBLEMA DE PROTEÇÃO E CONTROLE DE USO

Este capítulo apresenta os problema de proteção e controle de uso dos dispositivos móveis por parte do usuário, visando a maximização de sua experiência de utilização do seu dispositivo.

Dentre as várias técnicas para a solução deste problema, descrevemos em particular, a ativação de regras sensíveis ao contexto para a proteção com atenção no aspecto do consumo de energia da bateria do dispositivo.

3.1 Problema de Proteção e Controle de Uso de Dispositivos Móveis

Os dispositivos móveis tem problemas de proteção e controle de seu uso, onde o usuário pode ter seu dispositivo roubado ou o funcionamento do mesmo não satisfizer suas necessidades e/ou desejos.

Proteção sensível ao contexto é uma solução de proteção sensível ao contexto dinâmico e restringe o acesso a recursos (documentos, *e-mails*) e serviços (câmera, Internet, telefone, mensagens, dados do dispositivo móveis ou solicitar uma senha / padrão de acesso baseado em uma localização geográfica), com base numa política (conjunto de Regras Sensíveis ao Contexto) pré-definida e do contexto instantâneo do dispositivo. Pode fornecer uma melhor proteção para o conteúdo confidencial, dados do usuário e para garantir a integridade de vários serviços. Aplicações sensíveis ao contexto de localização, tais como a aplicação Locale (ou Tasker, mais completa e voltada inclusive para proteção) estão disponíveis no Android Market, mas estas aplicações exigem alta interação com o usuário na definição de regras de proteção e controle de utilização do dispositivo móvel.

No caso do Locale não é orientado a proteção e, sim, ao controle de uso baseado em localização geográfica.

O desafio inerente a estas soluções é automaticamente aprender e definir

políticas de maneira mais autônoma (por exemplo: reconhecimento de padrões, utilizando-se algoritmos adequados), de preferência sem forçar o usuário a pré-configurar toda política de proteção. Esforços de configuração das aplicações citadas acima podem ser classificadas como exigência de um usuário de nível médio ou avançado e uma modificação destes sistemas para simplificar a configuração utilizando mecanismos mais inteligentes, se faz necessária. (SHASBTAI, et al 2010).

O Controle de Acesso Sensível ao Contexto ou um *Context Aware Access Control* (conhecido pela sigla CAAC), pode limitar o acesso a dados privados, dependendo do contexto em que o dispositivo móvel se encontra, com base em sua localização, a rede de celular, se estiver conectado à Internet Wi-Fi entre muitos outros contextos. Esse mecanismo pode se defender contra uma variedade de ataques, dependendo da divulgação de informações em determinadas circunstâncias. Se o ataque ocorrer enquanto o dispositivo estiver num contexto que permite o acesso à informação e dados privados, o acesso será permitido e as informações divulgadas (SHASBTAI, et al 2010).

Controle de acesso sensível ao contexto pode lidar com redes protegidas e pode de fato ser visto como uma versão automatizada da abordagem de gerenciamento remoto dos dispositivos móveis. Sobre a detecção de um contexto que envolve uma conexão ativa com a rede protegida, o mecanismo da aplicação CAAC pode aumentar as medidas de proteção ativas no dispositivo. Tais medidas podem incluir a criptografia de conexão, autenticação e muito mais (SHASBTAI, et al 2010).

O desafio portanto, é proporcionar um mecanismo que possibilite a configuração de um maior número de regras sensíveis ao contexto de proteção e de controle de uso do dispositivo móvel, com o menor perda de desempenho e consumo de energia da bateria e, assim, maximizar a experiência do usuário no aspecto de proteção (defesa do dispositivo móvel contra ataques) no Android (RODRIGUEZ, CROWCROFT, 2011), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

A relação entre as regras de proteção sensíveis ao contexto adotam a seguinte

estratégia: o usuário do dispositivo móvel define que no contexto de uma determinada localização (posição geográfica), um determinado comportamento deve ser ativado através da solicitação de uma senha ou da desativação do padrão gráfico de acesso da tela.

Uma estratégia de proteção efetiva pode combinar mais de uma regra ativa de proteção: detecta a localização e aciona o comportamento esperado pelo usuário.

3.2 Mecanismos de Proteção Adicional de Controle de Uso

Até o momento, os trabalhos de investigação para impor políticas de proteção em dispositivos móveis, são de políticas consideradas inadequadas (por exemplo, permitir simplesmente que uma aplicação possa ser executada ou não).

Em Conti, Nguyen e Crispo (2010) foi apresentada uma proposta de utilização da aplicação chamada CREPE, um sistema capaz de aplicar políticas bem específicas, que variam enquanto uma aplicação é executada e também dependem do contexto de uso do dispositivo móvel.

Um contexto pode ser definido pelo estado de algumas variáveis (como localização, tempo, temperatura, ruído ou luz), a presença de outros dispositivos, a interação particular entre o usuário e o dispositivos, ou uma combinação destes.

A aplicação CREPE permite aplicar políticas (que são um conjunto de regras sensíveis ao contexto para uma única finalidade e entendendo uma regra como uma coleção de tarefas para atingir esta finalidade) relacionadas ao contexto quer pelo usuário, por terceiros de confiança ou de forma automática pelo sistema.

Dependendo da autorização dada a terceiros, pode-se definir uma política em um dispositivo em qualquer momento ou apenas quando ele está dentro de um contexto em particular, (dentro um edifício ou de um avião, exemplo) (CONTI,

NGUYEN, CRISPO, 2010).

Para enfrentar estes desafios, pode-se utilizar aplicações baseadas em um contexto de reconhecimento de modelo de controle de uso do dispositivo (ex: ConUCON ou CREPE), que aproveitam as informações de contexto para melhorar a proteção de dados, proteção de acesso e controlar o uso de recursos em um dispositivo móvel, estendendo o mecanismo padrão de proteção para implementar um quadro de execução de políticas que promovam uma maior proteção do Android).

Com este tipo de aplicação, adicionada ao sistema operacional para dispositivos móveis, os usuários serão capazes de empregar um refinado e flexível mecanismo de proteção para melhorar a proteção da privacidade e o controle de uso de recursos e informações de maneira adicional ao mecanismo de proteção padrão (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Nos países desenvolvidos há quase um assinante de telefone móvel para cada habitante, um a cada dois habitantes para os países em desenvolvimento e os riscos de proteção e ataques a computadores pessoais tradicionais, desde então, migraram para os dispositivos móveis. No mundo, há quase quatro bilhões de dispositivos móveis e estes dispositivos estão servindo cada vez mais e mais pessoas e organizações, como extensores e propensos substitutos dos computadores desktop (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Comparada com a proteção tradicional de computadores pessoais, a proteção de dispositivos móveis enfrenta ainda maiores desafios, pois muitos possuem características únicas como: personalização, mobilidade, serviços de pagamento eletrônico e recursos limitados (através de memorização de senhas, informações confidenciais, etc.), tendo potencializado o risco de perda e/ou roubo do dispositivo o que, conseqüentemente, expandiu à exposição da privacidade, bem como, o risco de roubo de informações confidenciais em um ambiente confidencial, (por exemplo uma reunião de negócios, uma conferência militar). (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Uma série de mecanismos de proteção, tais como UIDs, *label* de permissão, assinatura de aplicações e técnicas do tipo caixa de areia (mecanismo que dificultam o acesso da aplicação aos recursos não autorizados), foram adotados no Android para melhorar a sua proteção.

Com o melhor do conhecimento disponível e abordado até o momento, não há estudos existentes que combinem informações de contexto para fornecer de forma flexível, medidas de segurança / proteção / privacidade em dispositivos móveis, combinados com gerenciamento de energia e recursos de forma eficiente (RODRIGUEZ, CROWCROFT, 2011), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Descrevem-se como modelos de controle sensíveis ao contexto, tendo em mente as informações de contexto, como os dados espaciais e temporais durante a execução de uma aplicação, podendo estes serem capazes de suportar a proteção de dados flexíveis e o uso de recursos com restrições. (RODRIGUEZ, CROWCROFT, 2011), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Uma política relacionada ao contexto é uma política de proteção que exige a aplicação consciente do contexto do dispositivo móvel. Estendendo o mecanismo de proteção padrão existente no Android, para implementar um quadro de execução de políticas (conjunto de regras), uma política de proteção pode ser definida como uma declaração de que as partições dos estados do sistema, em um conjunto de estados de autorizados (seguros) e um conjunto de estados não-autorizados (inseguros). (RODRIGUEZ, CROWCROFT, 2011), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Estas novas técnicas combinadas oferecem diversos novos recursos de proteção, como permitir ao usuário conceder permissões de forma refinada e revogações destas e ainda, modificações em uma permissão na aplicação em tempo de execução (RODRIGUEZ, CROWCROFT, 2011), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Apesar do conceito que relaciona a proteção com o uso do contexto de controle de acesso não ser novo, os trabalhos aqui descritos trazem este conceito para o ambiente dos dispositivos móveis, onde as características desses dispositivos (alta mobilidade, constrangimentos de energia da bateria de dispositivos e capacidade de computação) tornam a proposta particularmente desafiadora (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

Ampliar o conjunto de tais contextos, não é possível. No entanto, a principal característica que os contextos têm, é que eles também podem ser automaticamente detectados e ativados pela aplicação de proteção pelo controle de uso baseado em contexto (por exemplo um novo contexto é detectado e ativado, quando o dispositivo entra em uma determinada área geográfica / região).

Destaca-se que neste trabalho, o conceito de contexto que difere das ativações simples e já existentes nos dispositivos como as opções "Modo de vôo" ou o "Modo silencioso", (que na verdade só podem ser ativados manualmente pelo usuário), podendo não apenas reforçar as políticas em tempo de execução, mas também permitir que terceiros de confiança (como agências autorizadas do governo), possam definir e ativar políticas através de um SMS com a semântica apropriada como visto em (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

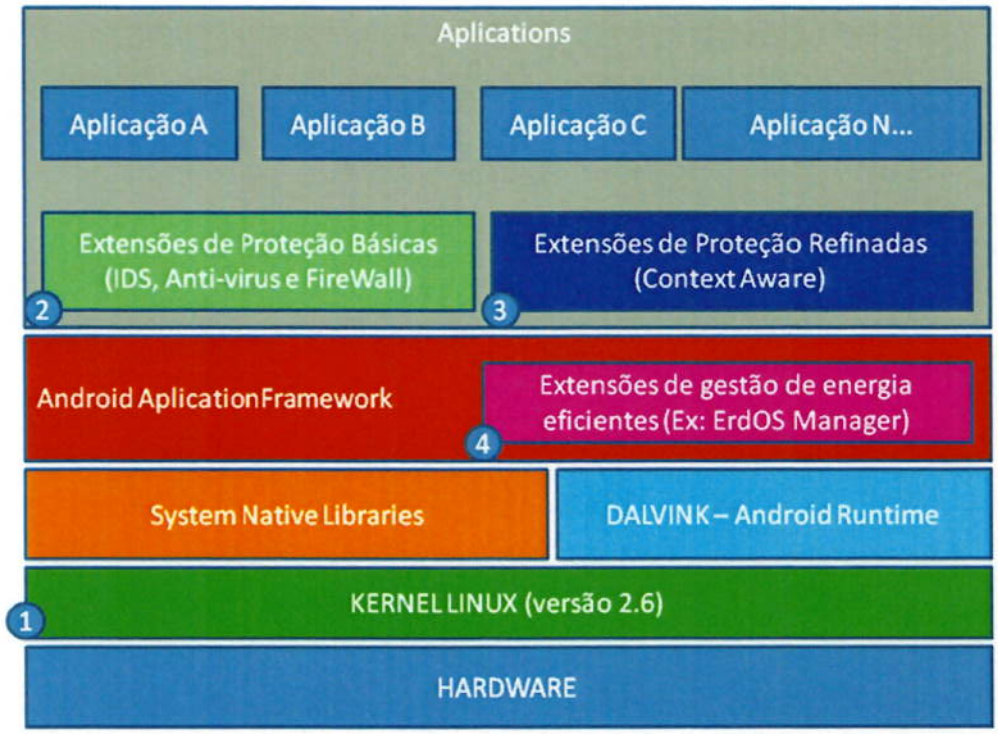
As soluções de sensíveis ao contexto implementadas (ConUCON ou CREPE) para resolver o problema de proteção contra ataques por controle de utilização de dispositivos móveis gozam de uma pequena sobrecarga, tanto em termos de tempo de processamento como em consumo de energia da bateria, em função da quantidade e/ou complexidade de regras sensíveis ao contexto e é gerenciável evidenciando assim, a sua viabilidade para solucionar o problema proposto.

Quando se utiliza um sistema operacional de gerenciamento de energia adicional (por exemplo, o Erdos) como na arquitetura mostrada na Figura 2, é possível adicionar uma quantidade/complexidade de regras sensíveis ao contexto maior para a proteção contra ataques no Android, com o mesmo

consumo de energia e não somente quando estas quantidades/complexidades de regras são mais simples, objetivando obter uma maior proteção do dispositivo móvel contra ataques em diversos contextos (RODRIGUEZ, CROWCROFT, 2011), (BAI, et al 2010), (CONTI, NGUYEN, CRISPO, 2010).

A Figura 2 é um complemento da Figura 1. Na Figura 1 mostrou-se a arquitetura e o mecanismo padrão de proteção do sistema operacional Android, na Figura 2 mostra-se as técnicas adicionais de proteção que podem ser implementadas nas camadas desta mesma arquitetura, com as camadas propostas para melhoraria na experiência do usuário na utilização do dispositivo móvel com foco em proteção, controle de utilização por regras sensíveis ao contexto e consumo de energia adequado.

Neste trabalho as técnicas mais relevantes são as Extensões de Proteção Refinadas (3) por regras sensíveis ao contexto (*Context-Aware*) e a Extensões de gerenciamento de energia eficientes (4) também por regras sensíveis ao contexto (*Context-Aware*).



Legenda:

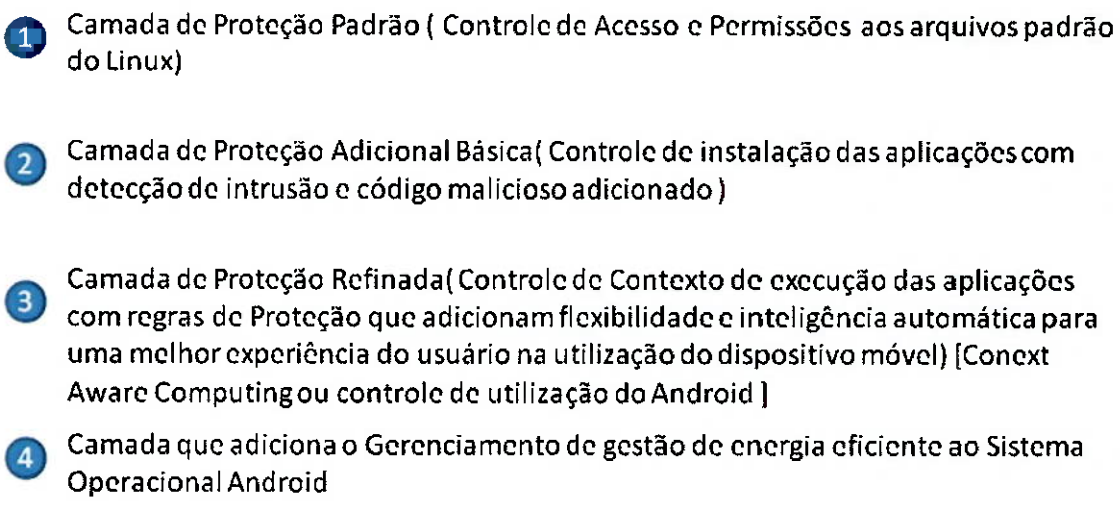


Figura 2. Arquitetura com as camadas propostas

3.3 Considerações do Capítulo

Neste capítulo foi descrito o problema e a necessidade de proteção e controle de uso adicional ao padrão do Android onde o usuário dos dispositivos móveis pode ter suas informações ou mesmo o próprio dispositivo roubados.

Ao adicionarmos proteção adicional ao Android através da técnica de ativação de regras sensíveis ao contexto é possível obter-se a melhor experiência de usuário desejada.

No próximo capítulo realizaremos o experimento de ativação e medição das regras sensíveis ao contexto em relação ao consumo de energia da bateria provocado e da proteção obtida. Isto dentro da perspectiva de uma melhor experiência do usuário e da solução do problema descrito acima.

4 PROPOSTA DE EXPERIMENTO E RESULTADOS

Este Capitulo tem como objetivo apresentar a comparação dos dois resultados e verificar os pontos comuns ou diferentes entre os mesmos.

Também foi feita uma análise da sobrecarga de energia provocada pela utilização de regras sensíveis ao contexto através da comparação entre uma pesquisa e um experimento proposto tendo como objetivo identificar os pontos que devem ser levados em consideração na utilização desta técnica.

4.1 Análise do Trabalho de Referência

Conti, Nguyen e Crispo (2010) descrevem os resultados experimentais do desempenho da utilização da ferramenta CREPE, avaliando a sobrecarga do consumo de energia que é uma das questões fundamentais para utilização de dispositivos móveis e também em relação a sobrecarga de processamento.

O dispositivo utilizado foi o *smartphone* HTC Magic. Foram identificadas duas características principais da utilização do CREPE que induz sobrecarga de energia e processamento comparada ao Android, sem a utilização de uma camada de regras de contexto: (i) ao chamar a PermissionChecker CREPE antes do Check Permissão do Android e (ii) ao determinar as mudanças de contexto por meio ContextInteractor. Também foi discutida a sobrecarga induzida por (i), observando-se que esta é gerada para cada pedido para o Android, que é interceptado por CREPE (CONTI, NGUYEN, CRISPO, 2010).

Foram realizados experimentos para entender a quantidade de sobrecarga induzida pela verificação de permissão do CREPE, tanto em termos de tempo, quanto em consumo de energia (CONTI, NGUYEN, CRISPO, 2010).

Quanto a sobrecarga de tempo que foi medido, (tempo induzido pelo verificado no CREPE, como mencionado anteriormente), essa verificação foi feita através de um acesso antes da verificação de permissão do Android. Foram medidos os intervalos de tempo entre o pedido de um recurso (aplicação ou sistema de serviço) e o momento em que o pedido é iniciado após ser verificado pelo

Android. Consideraram-se o pedido de acesso para aplicações e os serviços do sistema e não foram experimentadas as diferenças entre estas duas medidas.

Na verdade, ambos os pedidos foram processados da mesma maneira por CREPE que é uma aplicação de ativação de regras sensíveis ao contexto que adiciona flexibilidade e controle ao Andorid. Além disso, durante os experimentos foi notado que o recurso específico das regras, não influenciava os resultados (CONTI, NGUYEN, CRISPO, 2010).

Como a verificação de permissão é a ação mais comum de utilização do CREPE e, como já foi descrito, o consumo de energia sendo uma das questões mais importantes para a utilização dos dispositivos móveis, foi investigada a sobrecarga de energia induzida pela utilização de CREPE. Com isso pretendia-se verificar se mesmo com a ativação de um número maior de regras sensíveis no contexto, bem como, sua complexidade (em termos das ações realizadas pelas regras), este consumo de energia seria maior. Para investigar este aspecto, foi realizado um experimento com um cenário de alta demanda de energia e alto envolvimento de utilização do CREPE. Em particular, foi escrita uma aplicação que a cada dez minutos colocava um telefonema de 165 segundos (uma chamada gratuita para um Serviço automático) no dispositivo. Esta aplicação funcionou por duas horas, com o inicio da experiência com a bateria totalmente carregada (Bateria em 100%) (CONTI, NGUYEN, CRISPO, 2010).

Foram realizados 10 experimentos para cada um dos dois casos seguintes:
- Telefone rodando o Android, sem adição de regras de contexto;
- Utilizando o CREPE no dispositivo funcionando e verificando a evolução em função do número de regras.

No caso da utilização do CREPE, a experiência foi repetida para um número diferente de regras ativas. Foi verificado que não é significativa qualquer diferença para os recursos específicos das regras (para que finalidade elas são). Os resultados dos experimentos são mostrados na Figura 3.

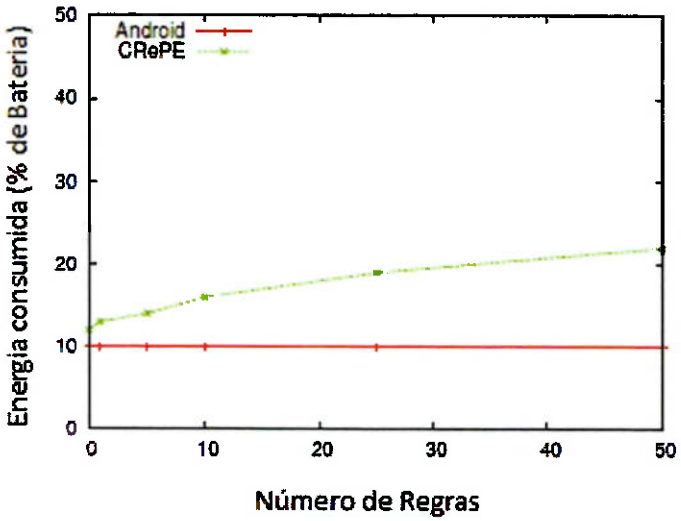


Fig. 3. Consumo de Energia da Bateria

Figura 3. Gráfico com o resultado do consumo de energia da bateria do dispositivo móvel com a adição do número de regras sensíveis ao contexto.

Como esperado, na Figura 3, observa-se que o consumo de energia com a utilização do CREPE (linha verde) é maior que no Android original (linha vermelha). Foi verificado que devido à verificação de permissão do CREPE o consumo de energia é maior. Durante o experimento de utilização do CREPE foram realizadas 2.721 verificações de permissão de chamadas por 19 diferentes recursos principalmente para permissão para o dispositivo de alimentação ou consumo de energia da bateria, 1407 vezes e permissão (AKE LOCK, 588 vezes) (CONTI, NGUYEN, CRISPO, 2010).

Conforme as expectativas dos autores, o consumo de energia aumenta com a utilização do CREPE enquanto se aumenta o número de regras ativas.

A sobrecarga de energia induzida por CREPE é cerca de 50% maior do que o consumo padrão do Android quando 10 regras são definidas, e, aumenta em 100% se 50 regras sensíveis ao contexto são definidas. Verificou-se que o consumo de energia não aumenta linearmente com o número de regras ativas devido a alguma ação básica de CREPE que não depende do número e do conjunto de regras (CONTI, NGUYEN, CRISPO, 2010).

Durante o experimento, os autores também variaram o conjunto de regras ativas para um determinado tamanho do conjunto regras proposto e observaram que os resultados não dependem das regras que estão ativas (CONTI, NGUYEN, CRISPO, 2010).

Na verdade, para cada invocação de CREPE que verifica a permissão solicitada junto ao dispositivo e comparando-a com cada uma das regras ativas, o *overhead* depende apenas do número de regras e não da sua natureza e característica (CONTI, NGUYEN, CRISPO, 2010).

A partir deste experimento, pode-se concluir que o consumo de energia das permissões verificadas não é desprezível. No entanto, foi salientado que a utilização de 50 regras ativas não é comum e, mesmo neste caso, consumindo muita energia, fica demonstrado que a solução CREPE ainda é viável (CONTI, NGUYEN, CRISPO, 2010).

Foi também sublinhado que esta é a primeira implementação de CREPE e que não foi destinada uma atenção especial para otimizações (CONTI, NGUYEN, CRISPO, 2010).

Quanto à sobrecarga de energia, esta depende principalmente de quanto o sistema é sensível às mudanças de contexto através das regras ativas e, na verdade, quanto mais cedo se deseja detectar um contexto, com maior freqüência a verificação do contexto ocorrerá. Portanto, maior será o *overhead* (CONTI, NGUYEN, CRISPO, 2010).

Isto é verdadeiro para as características de contexto que precisam ser verificadas ativamente pelo dispositivo móvel, como as coordenadas por GPS. Enquanto isso, não vale para um contexto explicitamente definido pelo usuário, como exemplo, uma ação quando o dispositivo receber um SMS ou por terceiros (através de mensagens enviadas que provocam uma ação) (CONTI, NGUYEN, CRISPO, 2010).

A abordagem mais simples foi considerar uma monitoração contínua para a detecção de mudanças de contexto. Nos experimentos para avaliar a atual

implementação de CREPE realizados na pesquisa foi utilizado o sinal de GPS em sondagens que foram realizadas sem qualquer contexto ativo, apenas para medir a sobrecarga do Polling GPS. Considerou-se o consumo de energia observado em 5 consecutivas horas, começando com uma bateria totalmente carregada para monitorar este consumo (CONTI, NGUYEN, CRISPO, 2010).

Observou-se que se os pedidos CREPE da posição atual para o dispositivo GPS a cada 5 minutos fossem configurados, o nível da bateria diminui em 48%, enquanto verificando-se a posição pelo GPS a cada 15 minutos consumiria apenas11% da bateria (CONTI, NGUYEN, CRISPO, 2010).

Na verdade, no problema abordado pelo solução CREPE, verifica-se que otimizações são possíveis a partir deste ponto de vista, como por exemplo, o local que é verificado com freqüência pode ter sua freqüência diminuída, enquanto o valor de uma variável de interesse (como a localização) é o valor para o qual o contexto é ativado (CONTI, NGUYEN, CRISPO, 2010).

Foi observado como a implementação atual de CREPE tem razoável quantidade de sobrecarga, do ponto de vista energético (Consumo de energia da Bateria) (CONTI, NGUYEN, CRISPO, 2010).

Em função do descrito acima, algumas otimizações são possíveis em relação a implementação da versão atual do CREPE avaliada no trabalho de Conti, Nguyen e Crispo (2010), como a verificação de localização pode ser otimizada aproveitando outras informações por meio do provedor da operadora de telefonia ou interfaces Wi-Fi disponíveis (como o roteador de uma rede local) ou ainda, com o GPS sendo acionado de forma mais esporádica.

4.2 Experimento Proposto

Com o objetivo de analisar os resultados obtidos no trabalho de Conti, Nguyen e Crispo (2010), é proposto um experimento com a ferramenta Tasker baixada de Android Market (2012), com o intuito de verificar o consumo de energia da bateria com utilização de regras sensíveis ao contexto num dispositivo móvel.

Este experimento foi realizado com o dispositivo móvel modelo Sansung Galaxy 5, conforme a Figura 4.

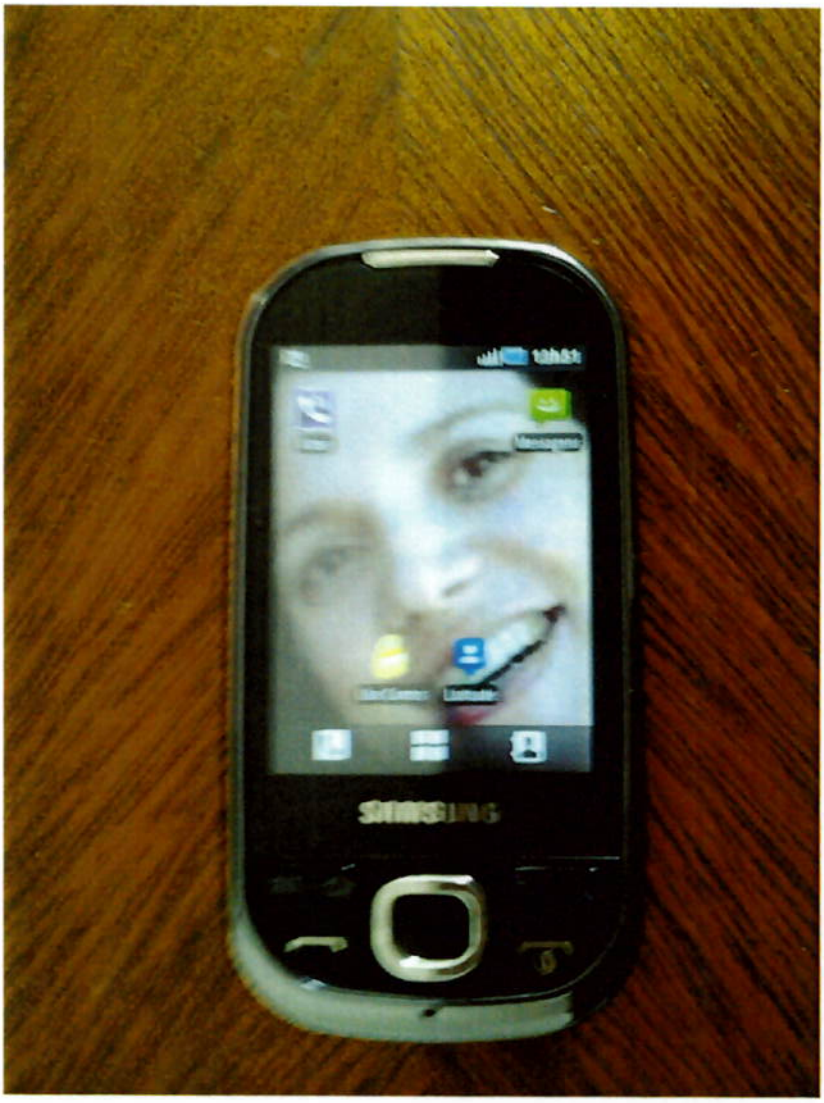


Figura 4. Foto do dispositivo utilizado no laboratório - Sansung Galaxy 5 com Android 2.2

4.2.1 Planejamento do Experimento

Como planejado o experimento foi preparado para medir a variação de consumo de energia da bateria do dispositivo ao realizar uma medida sem a adição de nenhuma regra de contexto e outras quatro medições com adição incremental subseqüente de 03, 06, 01 regra especial com alta utilização de GPS e mais 04 regras de contexto.

Proposta

Os testes são realizados rigorosamente nas mesmas condições: dispositivo com 100% de energia na bateria no inicio, as utilizações e medições realizadas nos mesmos períodos de tempo e a energia da bateria a 0% no final. O teste seguinte seguiu exatamente o mesmo processo.

Preparação do ambiente do experimento

Foi utilizado a aplicação One-Touch (Android Market – aplicação open source), Figura 5, para a verificação de quando a carga de energia do dispositivo estava em 100% no inicio de cada teste, Figura 6, e a 0% (a leitura passa de 2% para 0% muito rapidamente) no seu final Figura 7.

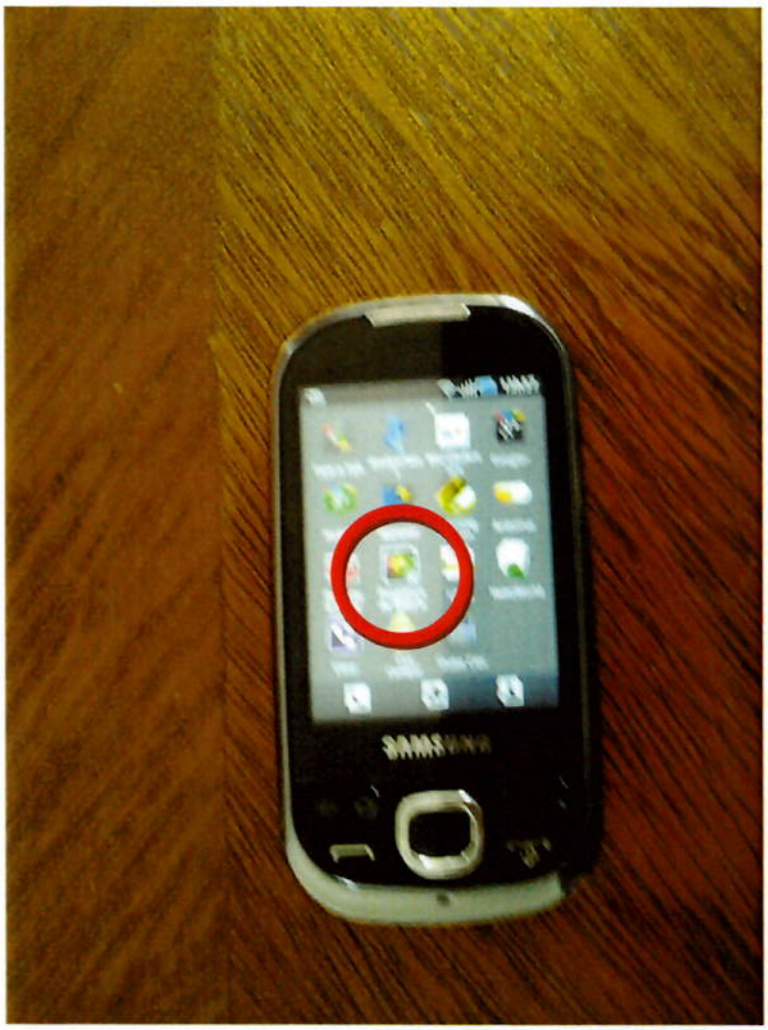


Figura 5. Foto aplicação One-Touch

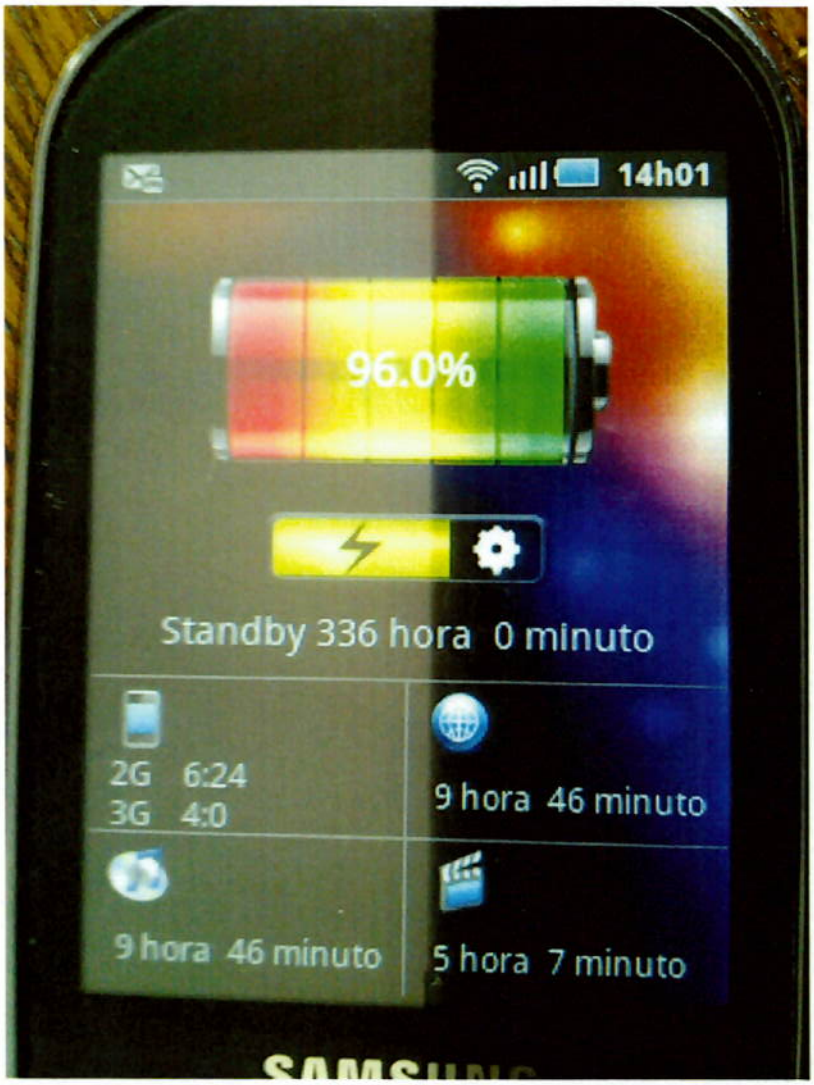


Figura 6. Dispositivo com 100% de Bateria

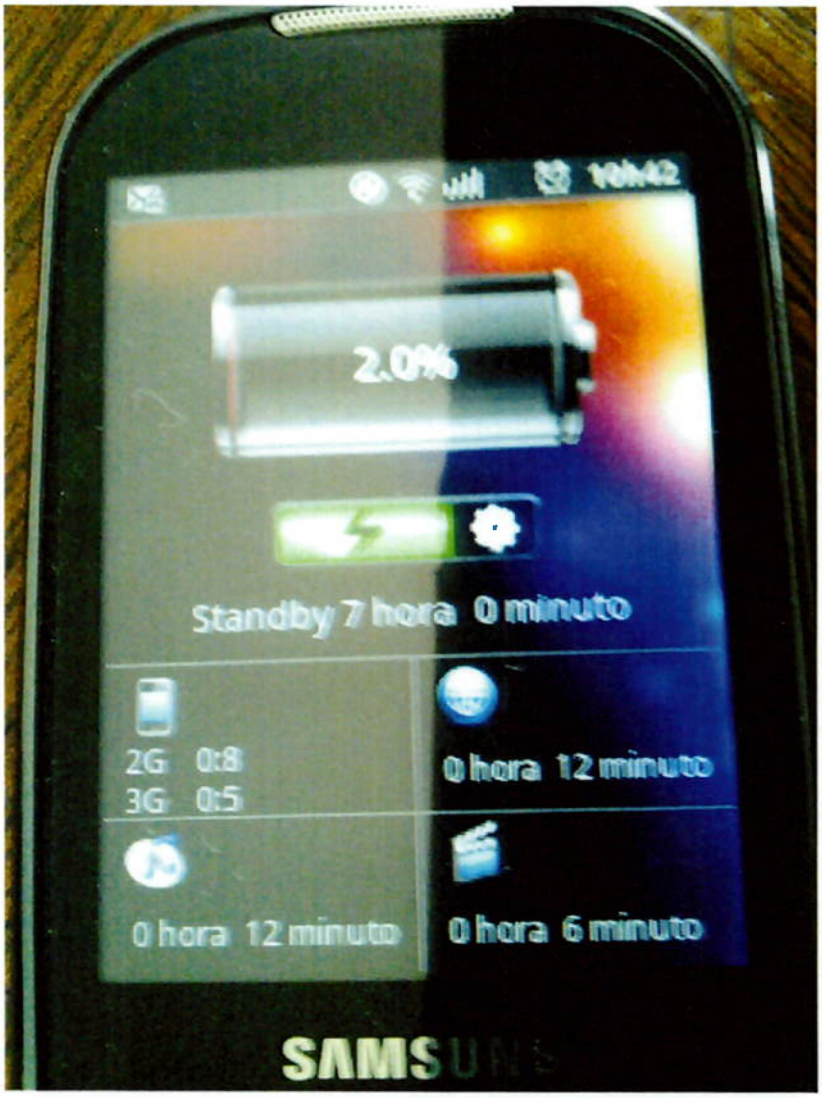


Figura 7. Dispositivo com 0% de Bateria

Para implementação das regras sensíveis ao contexto foi utilizado a aplicação Tasker (Android Market – aplicação paga) Figura 8.

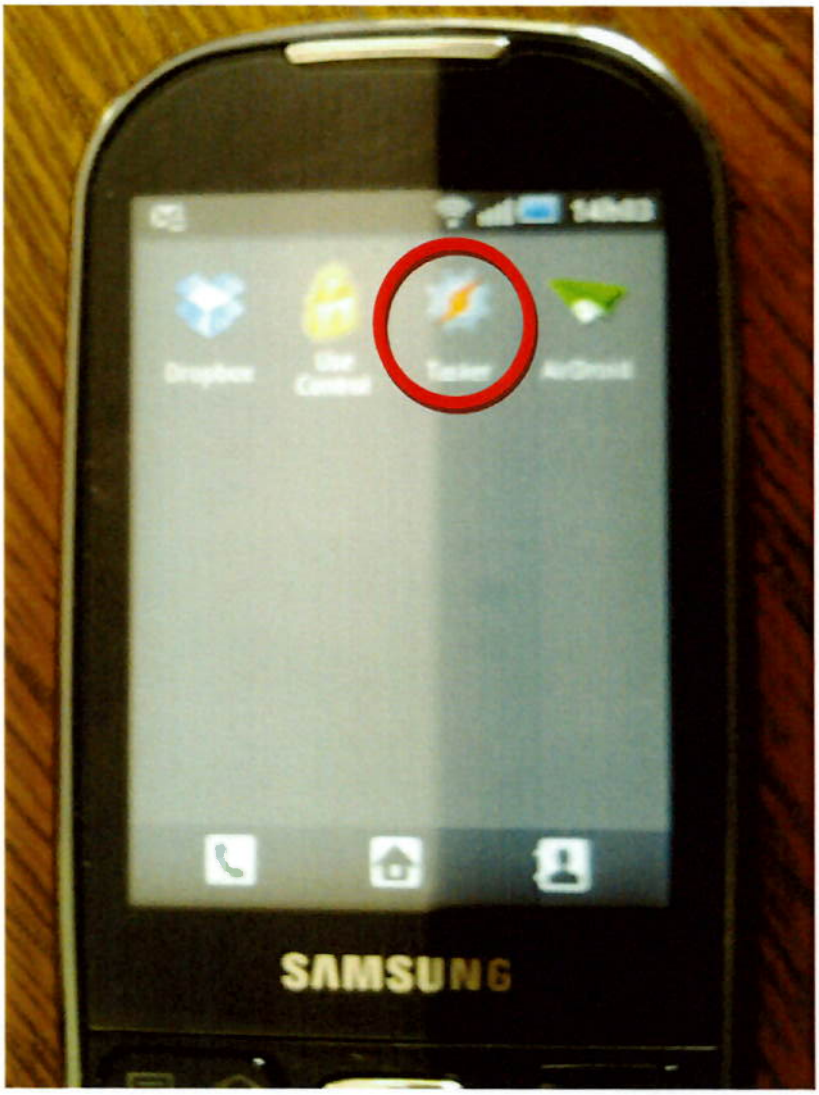


Figura 8. Foto da aplicação Tasker

Em todos os testes a ativação das regras, pode ser verificada pelo ícone no canto superior esquerdo do dispositivo como mostra a Figura 9.

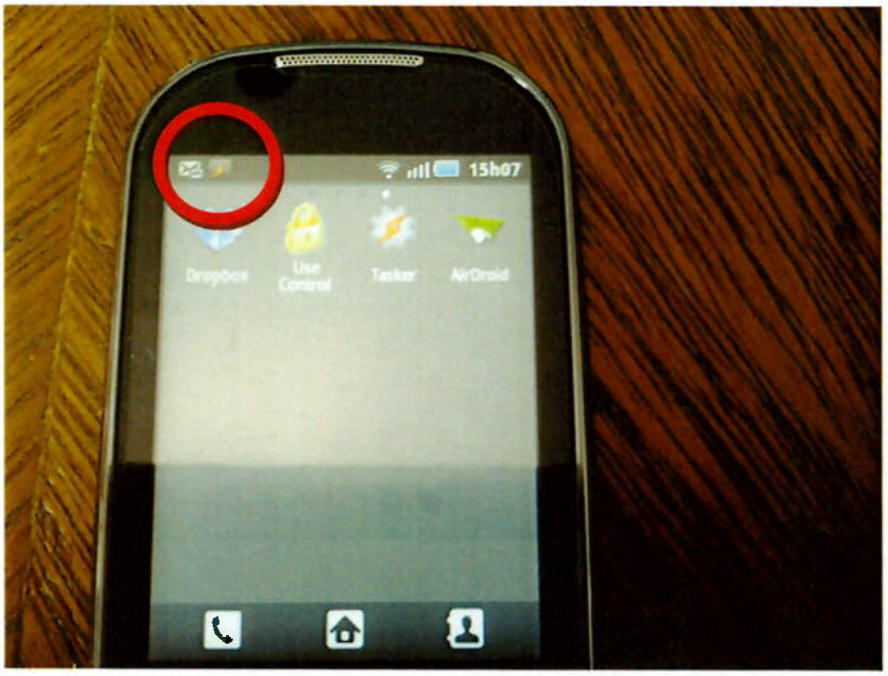


Figura 9. Foto regras ativas

No final dos testes as regras foram desativadas como mostra as Figuras 10 e 11.

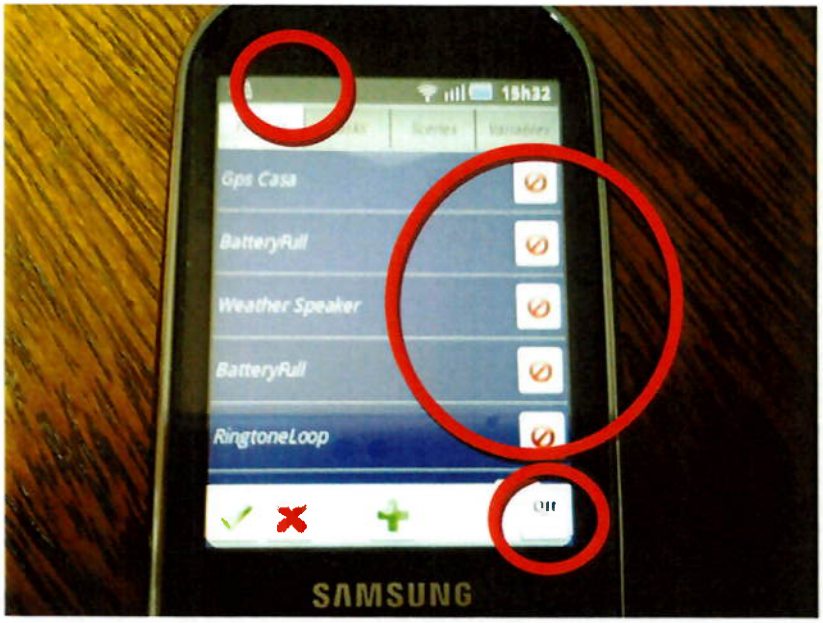


Figura 10. Foto com todas as regras sensíveis ao contexto sendo desativadas.

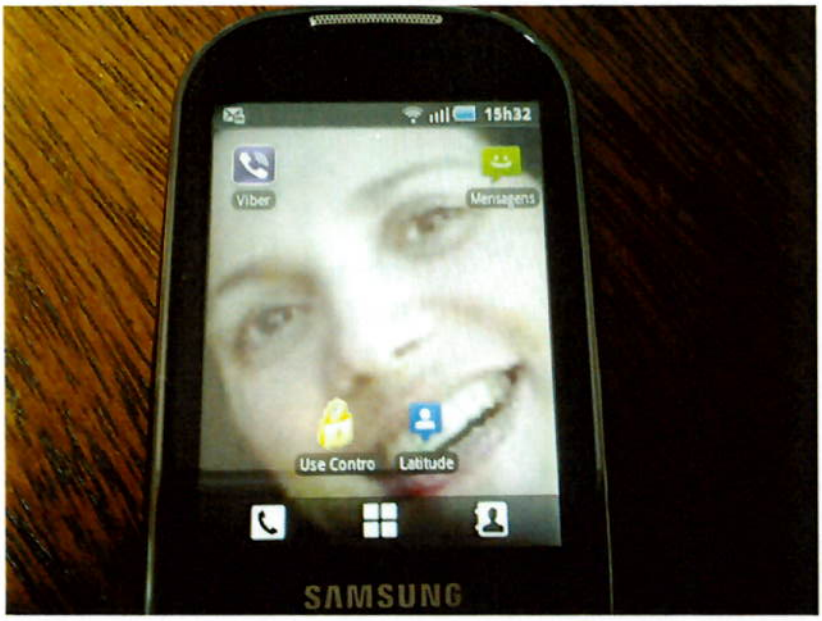


Figura 11. Foto com dispositivo sem nenhuma regra ativa após o laboratório.

Lista de Regras Utilizadas nos Testes

Estas regras sensíveis ao contexto foram escolhidas dentre várias outras disponíveis no site de regras sensíveis ao contexto de exemplo da ferramenta Tasker da empresa americana Crafty Apps em Tasker (2012). Com exceção da regra especial com grande utilização do GPS que foi modificada baseada em uma já existente (Wifi Automator) todas as demais foram utilizadas sem modificação e estão disponíveis para download no site citado. O critério para a escolha das regras foi definido pelas regras que não envolviam uma complexidade para serem ativadas e verificadas nos testes propostos e que, apesar disto, consumiam algum recurso do dispositivo podendo diminuir o nível de energia da bateria.

Regras Para o Teste 2

- ✓ Battery Full
  - o Descrição: Esta regra de contexto fornece uma notificação uma vez que a sua bateria está cheia e não vai repetir até que você desconecte e a conecte novamente.

- ✓ Weather Speaker
    - Descrição: Irá criar uma regra de contexto que vai verificar as condições meteorológicas diárias nos seguintes horários: às 7:05 am e 7: 08 am e informar.

- ✓ Ringtone Loop
    - Descrição: Esta regra de contexto permite que você mude os toques a cada nova chamada recebida. É assumido que temos apenas três toques para utilização.

Regras Para o Teste 3

- ✓ Headphones Plugged In
    - Descrição: Esta regra aciona a aplicação de áudio quando o fone é inserido.
    - Se você quiser retornar à tela inicial quando fone é removido, você precisa adicionar uma regra de contexto de tarefa Exit que é executado quando há saídas de contexto ou seja, quando o fone de ouvido é removido.

    - Estas duas regras abaixo sensíveis ao contexto, quando ativadas vão continuar a repetir o nome do chamador (Em uma ligação telefônica) até que o telefone seja atendido ou o chamador desista.

- ✓ PhoneAnswered
    - Descrição: Esta regra de contexto detecta o telefone que está sendo respondido com um evento offhook de telefone que para o laço (loop) do discurso.

- ✓ PhoneRinging
    - Descrição: Esta regra de contexto detecta o início e o fim de uma chamada com o estado de chamada recebida.

Regras Para o Teste 4

- ✓ GPS Casa (Adaptação da regra de contexto Wifi Automator)
    - Descrição: Esta regra detecta a sua localização pelo GPS (Verificada a cada 15 minutos) num raio de 200 metros, se você esta em casa e abre o dispositivo móvel não é solicitado o padrão de senha para a utilização enquanto se permanece em casa. Se você não estiver em casa o dispositivo volta a solicitar o padrão de senha para sua utilização. O objetivo desta regra de contexto é demonstrar o maior consumo de energia pela intensa utilização do GPS o que NÃO ajuda na economia da bateria.

Regras Para o Teste 5

- ✓ Audio Level for Different Daily Times
    - Descrição: Estas 04 regras sensíveis ao contexto ajustam automaticamente o volume de áudio para momentos diferentes do dia, como Manhã, Tarde e Noite:
    - Audio Levels Evening
    - Audio Levels Morning
    - Audio Levels Night
    - Audio Levels Work

4.2.2 Execução do Experimento

Os testes obedeceram a seqüência abaixo:

Teste 1 – Verificação do consumo de energia sem a ativação de regras de contexto;

Teste 2 – Verificação do consumo de energia com a ativação de 03 regras de contexto;

Teste 3 – Verificação do consumo de energia com a ativação de mais 03 regras sensíveis ao contexto, em adição as regras anteriores;

Teste 4 – Verificação do consumo de energia com a ativação de mais 01 regra de contexto (Regra com alto consumo do sinal do GPS) em adição as regras anteriores;

Teste 5 – Verificação do consumo de energia com a ativação de mais 04 regras sensíveis ao contexto, em adição as regras anteriores.

O primeiro teste foi feito carregando a energia da bateria do dispositivo em 100% e executamos as mesmas utilizações (Uso do dispositivo móvel) descrita abaixo, em ambiente controlado, medindo e verificando quantas horas iria durar a energia da bateria até o seu termino em 0%.

Após este primeiro teste foi novamente carregada a bateria em 100% e três regras descritas abaixo foram adicionadas e novamente verificamos quantas horas durou a energia da bateria até o seu termino em 0%.

Para estas medições usou-se a aplicação de verificação do nível de energia da Bateria One-Touch como nas Figuras 5, 6 e 7. Sucessivamente, os próximos testes utilizaram mais 06 regras, 01 regra especial com grande utilização do GPS do dispositivo e finalmente mais 04 regras, sempre de forma incremental (Conservando as regras anteriores ativadas e ativando as subseqüentes), sempre com a mesma utilização descrita abaixo.

Um Chip pré-pago da operadora TIM foi adiquirido e utilizado para proporcionar um ambiente controlado durante os testes, o número de telefone foi ativado e utilizado "exclusivamente" para este fim.

A bateria do dispositivo foi carregada totalmente, descarregada (0%) e novamente carregada (100%) e utilizada até seu termino para cada um dos seguintes testes (ou medições):

Execução do Teste 1

O dispositivo, sem adição de regra sensíveis ao contexto, sofreu as seguintes utilizações:

Com 7 horas após o inicio da carga de 100% da bateria (sempre no mesmo horário) uma solicitação 3G, uma ligação de 30 segundos (sempre para o

mesmo telefone), receber uma ligação (sempre do mesmo telefone), enviar um SMS (sempre a mesma mensagem e para o mesmo telefone) e ligar o fone de ouvido, reproduzindo 30 segundos de áudio (sempre o mesmo áudio) e a mesma utilização descrita acima, 7 horas depois da primeira utilização. Entre estas utilizações o dispositivo permaneceu em repouso.

Neste teste nenhuma regra sensível ao contexto foi ativada no dispositivo como mostrado na Figura 12.

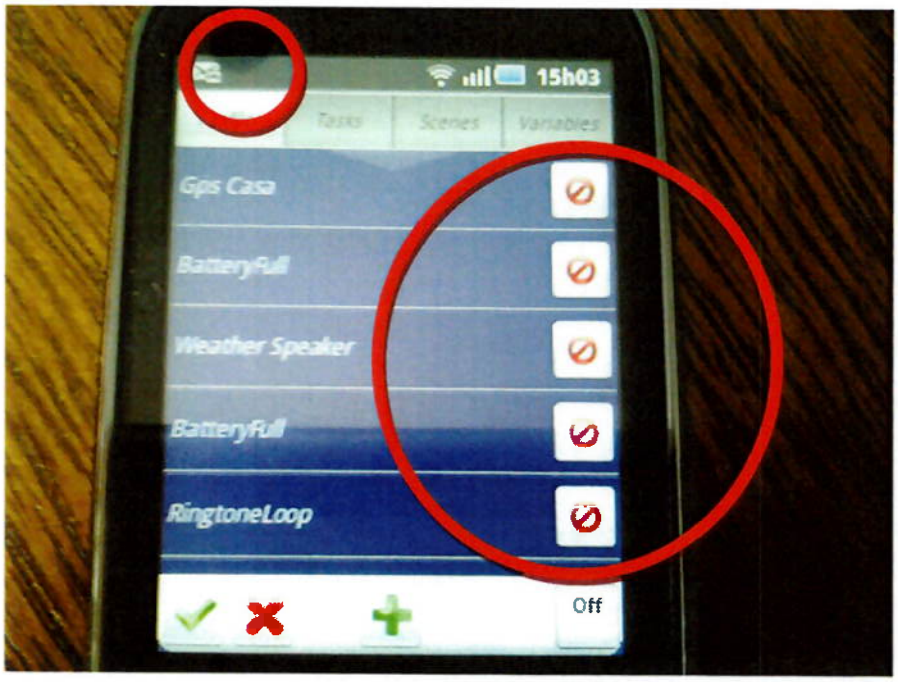


Figura 12. Foto sem a ativação de regra sensível ao contexto

Fim do teste: A duração da bateria foi de 23 horas.

Execução do Teste 2

Adição de três regras sensíveis ao contexto (Regras para o Teste 2) com as seguintes utilizações propostas:

Com 7 horas após o inicio da carga de 100% da bateria (sempre no mesmo horário) uma solicitação 3G, uma ligação de 30 segundos (sempre para o mesmo telefone), receber uma ligação (sempre do mesmo telefone), enviar um SMS (sempre a mesma mensagem e para o mesmo telefone) e ligar o fone de ouvido, reproduzindo 30 segundos de áudio (sempre o mesmo áudio). Executar

a mesma utilização descrita acima, 7 horas depois da primeira utilização. Entre estas utilizações o dispositivo permaneceu em repouso.

A ativação das regras do teste acima é ilustrada pelas Figuras 13.

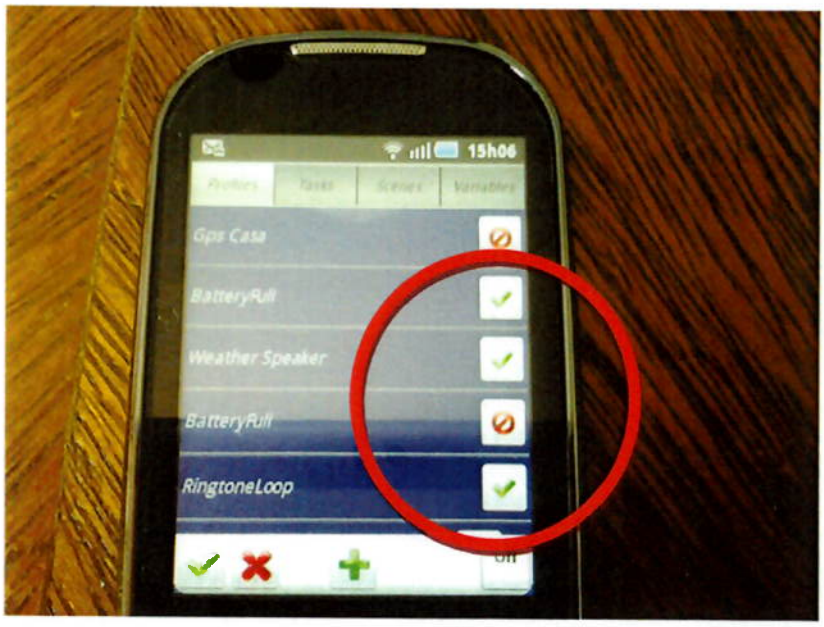


Figura 13. Foto Ativação das regras para o Teste 2

Fim do teste: A duração da bateria foi de 23 horas como no teste anterior.

Execução do Teste 3

Acrescentam-se mais três regras de contexto (Regras para o Teste 3) para as mesmas utilizações propostas.

Com 7 horas após o inicio da carga de 100% da bateria (sempre no mesmo horário) uma solicitação 3G, uma ligação de 30 segundos (sempre para o mesmo telefone), receber uma ligação (sempre do mesmo telefone), enviar um SMS (sempre a mesma mensagem e para o mesmo telefone) e ligar o fone de ouvido,reproduzindo 30 segundos de áudio (sempre o mesmo áudio). Executar a mesma utilização descrita acima, 7 horas depois da primeira utilização. Entre estas utilizações o dispositivo permaneceu em repouso.

A ativação das regras do teste acima é ilustrada pelas Figuras 14.

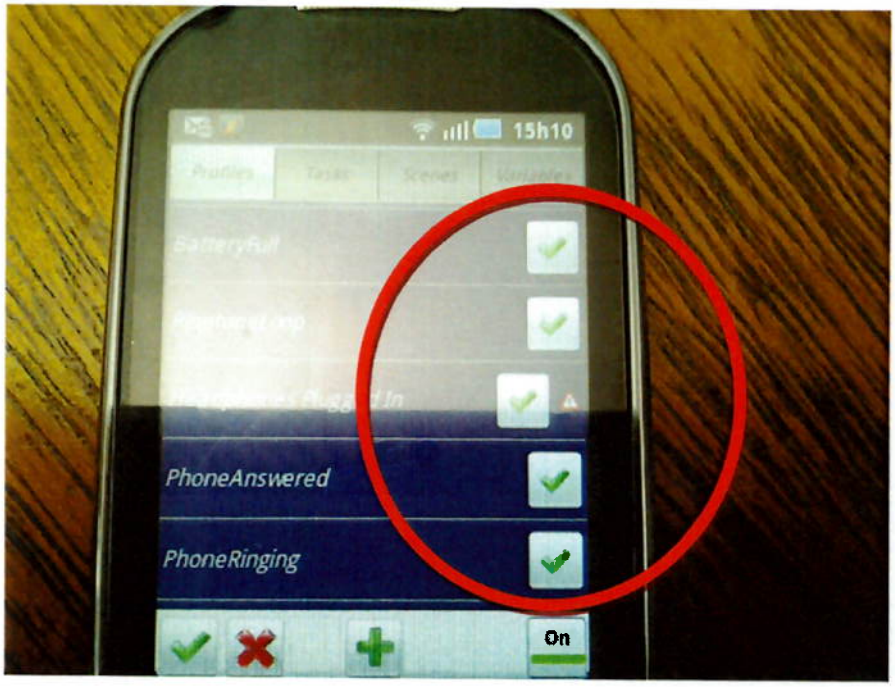


Figura 14. Foto Ativação das regras para o Teste 3

Fim do teste: A duração da bateria foi de 21 horas.

Execução do Teste 4

Acrescenta-se mais uma regra de contexto (Regras para o Teste 4) para as mesmas utilizações propostas no terceiro teste.

Foi definida e configurada uma ação onde o raio a partir da localização definida como "GPS casa" em metros, define quando é acionado o teclado virtual e senha. Fixado em 200 metros, ou seja, ao sairmos da área de 200 metros do local (que é verificado a cada 15 minutos pelo dispositivo) definido com "GPS casa" o dispositivo solicita a senha através do teclado virtual. Por padrão esta regra acionar o GPS a cada 15 minutos para verificar a localização. A posição da localização geográfica "GPS casa", obtida através do GPS e não de uma rede WiFi (Botão Net) foi alterada como ilustrado nas Figuras 15, 16, 17, 18, 19, 20, 21,22, 23, 24, 25, 26, 27, 28, 29 e 30.

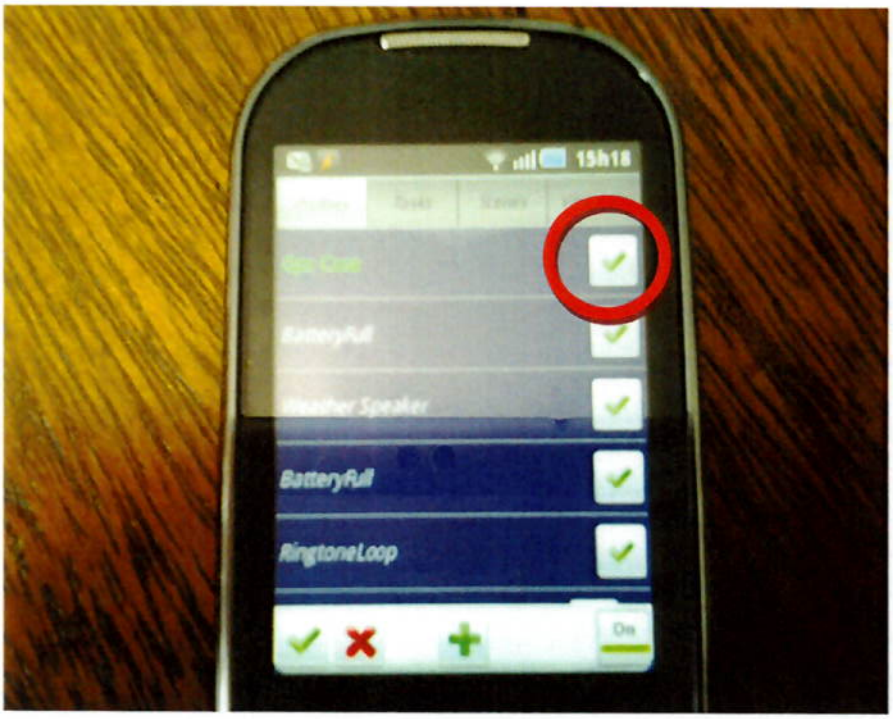


Figura 15. Ativação das regras para o Teste 4

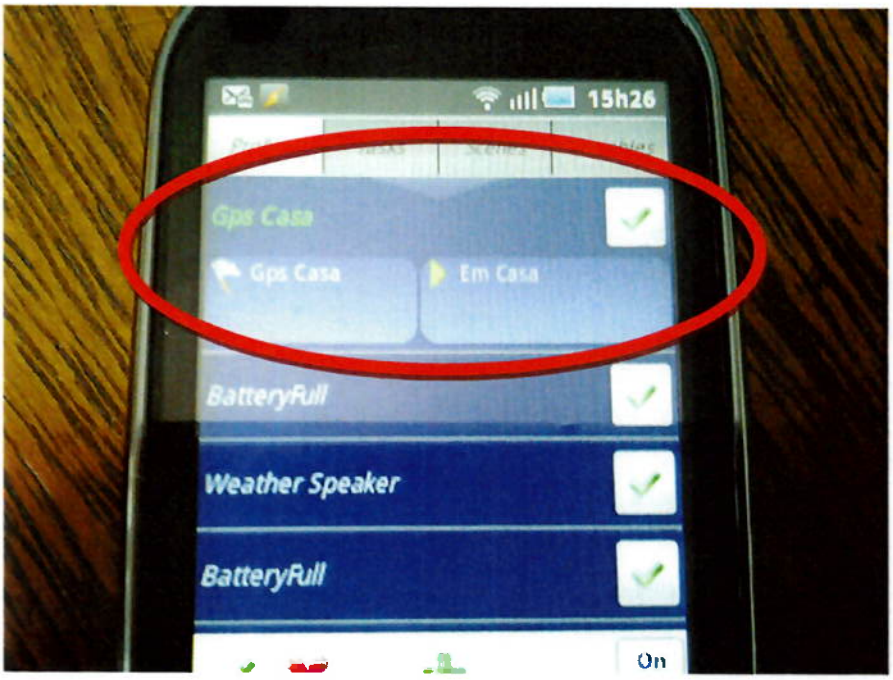


Figura 16. Foto Configurar - Adicionar e selecionar a regra.

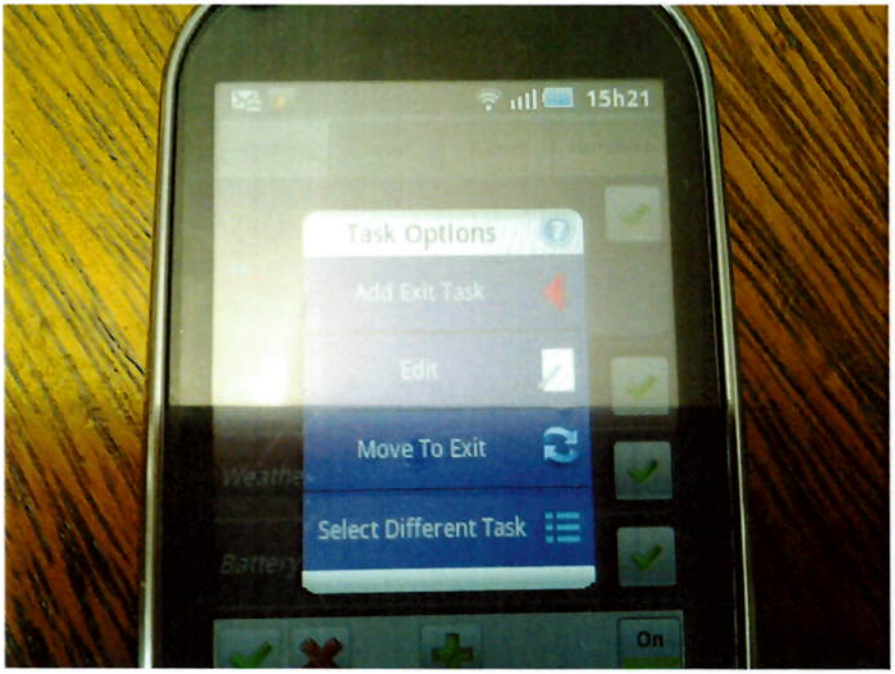


Figura 17.Foto Configurar - Editar a regra.

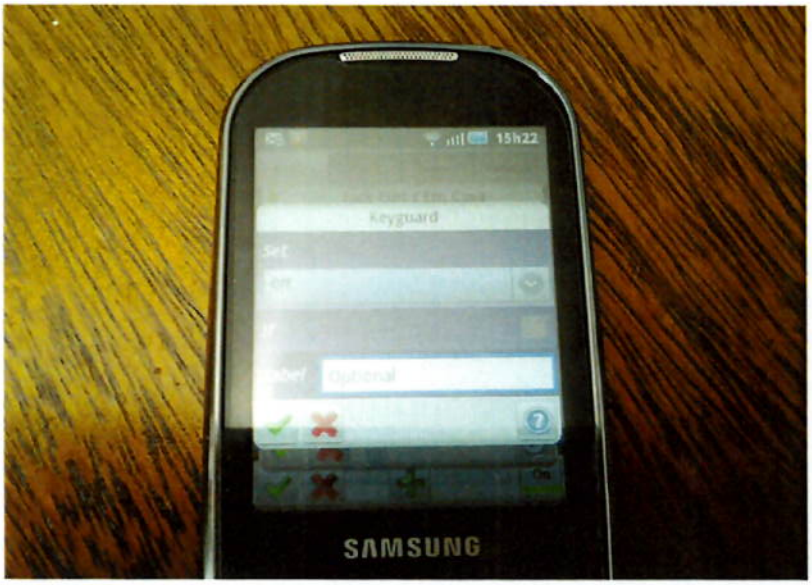


Figura 18. Foto Configurar - Definir a ação de controle de acesso da regra.

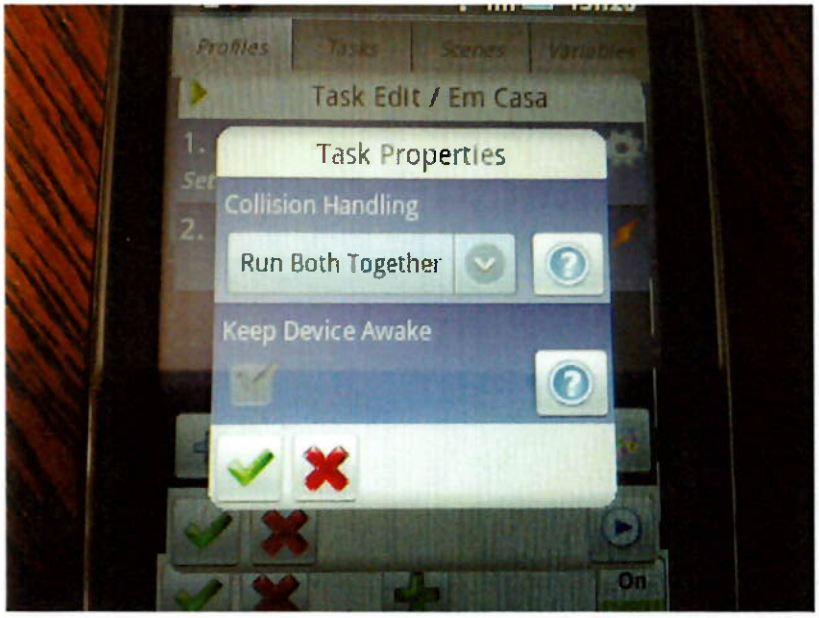


Figura 19 Foto Configurar - Definir a ação de controle de acesso e propriedades da regra.

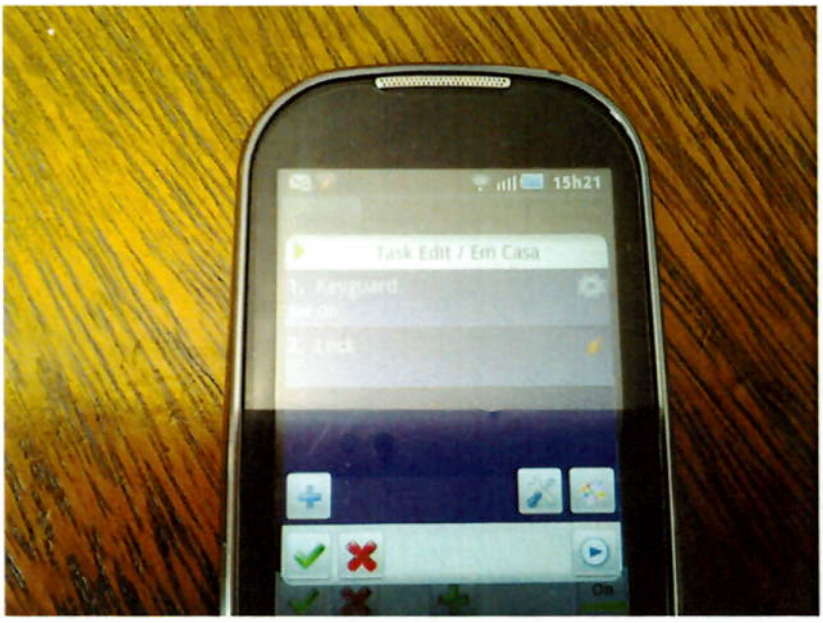


Figura 20. Foto Configurar - Definir a ação de controle de acesso e propriedades de teclado na tela da regra.

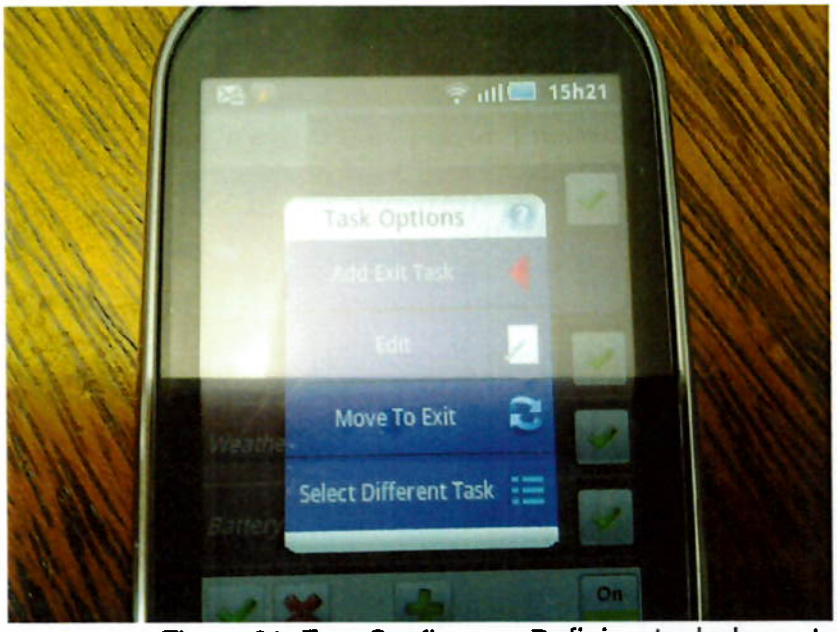


Figura 21. Foto Configurar - Definir o teclado na tela da regra

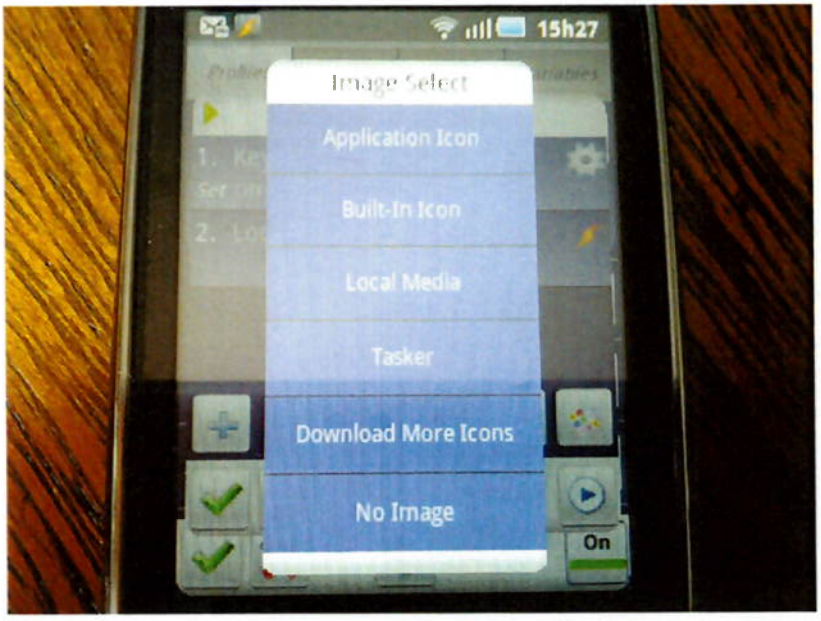


Figura 22. Foto Configurar - Definindo a imagem do teclado virtual.

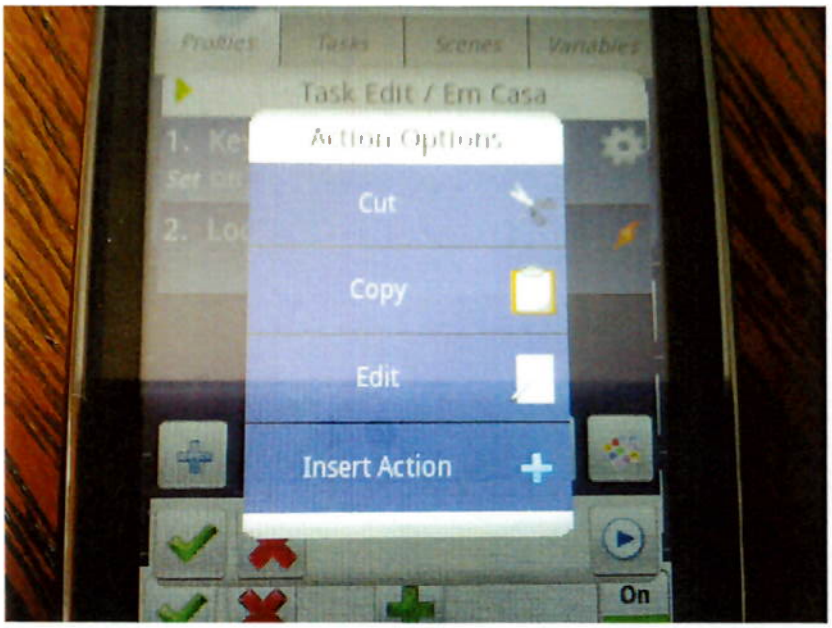


Figura 23. Foto Configurar - Definindo as ações do teclado virtual

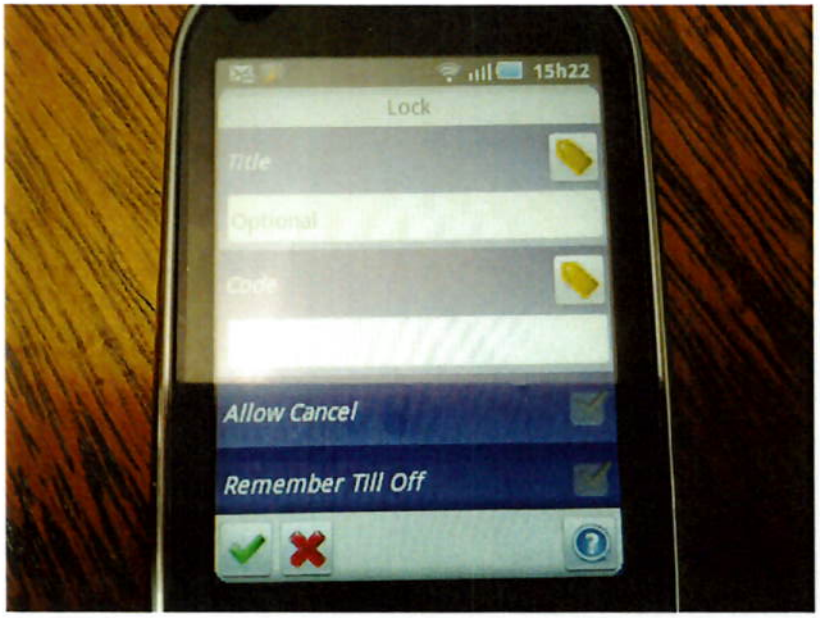


Figura 24. Foto Configurar - Definindo a imagem do teclado virtual e suas ações (Detalhe).

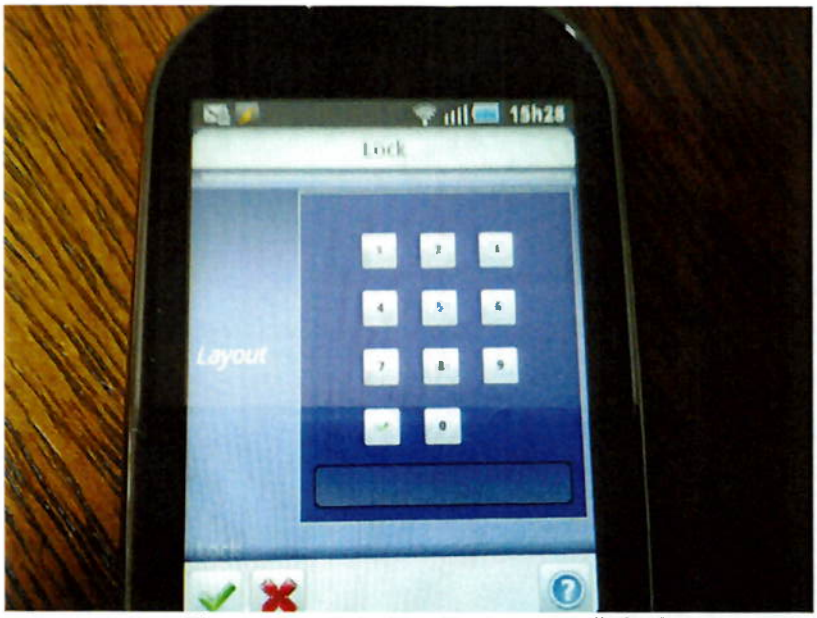


Figura 25. Foto Configurar - Definindo a senha de acesso

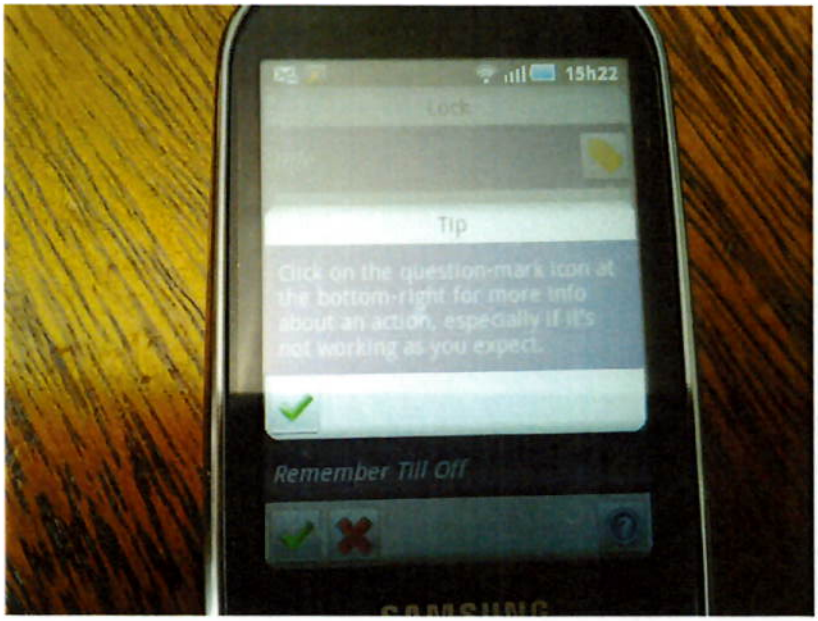


Figura 26. Foto Configurar - Ações adicionais possíveis.

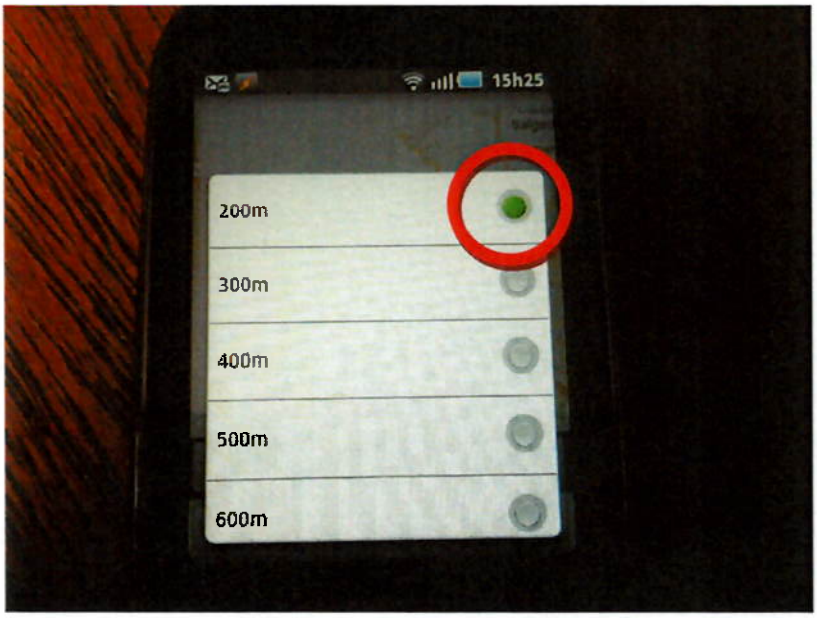


Figura 27. Foto Configurar - Área de 200 m do local definido como "GPS casa"

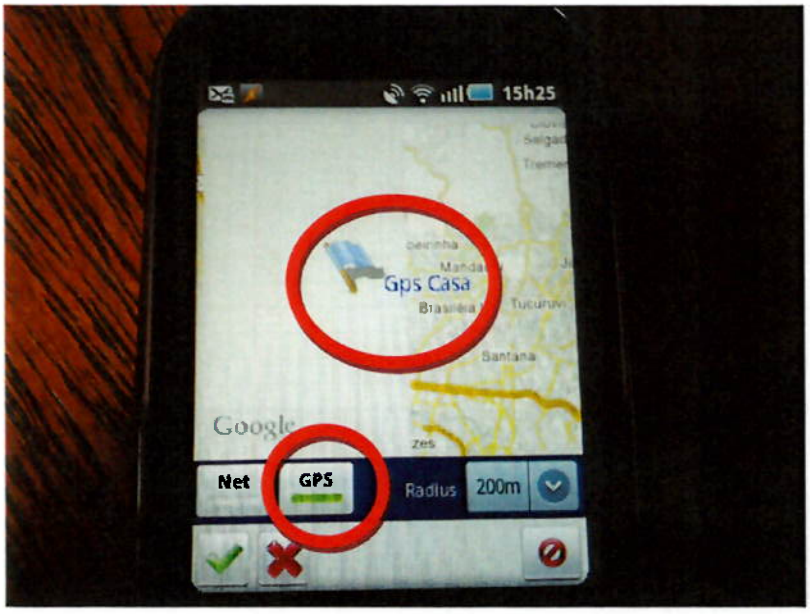


Figura 28. Foto Configurar - Seleção do local por GPS

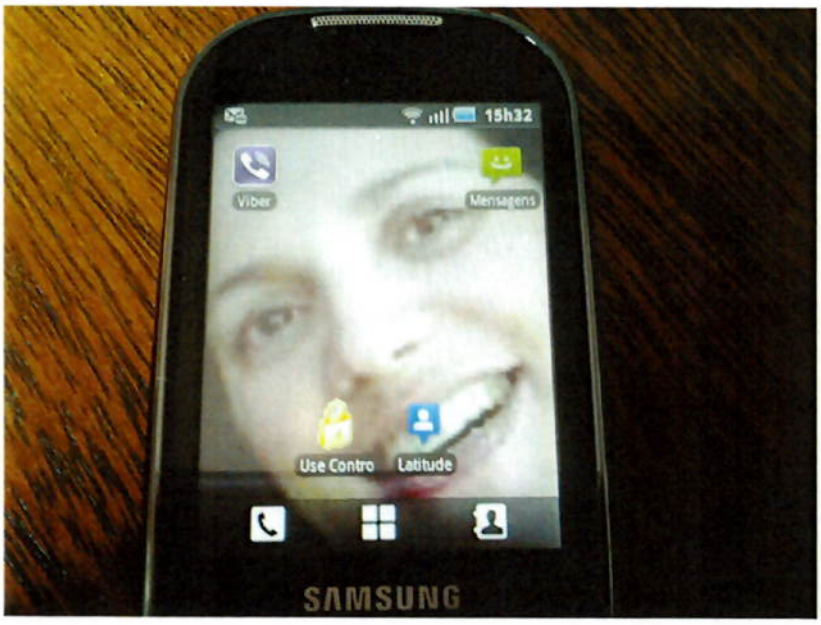


Figura 29. Foto quando o dispositivo se encontra no local "GPS casa"

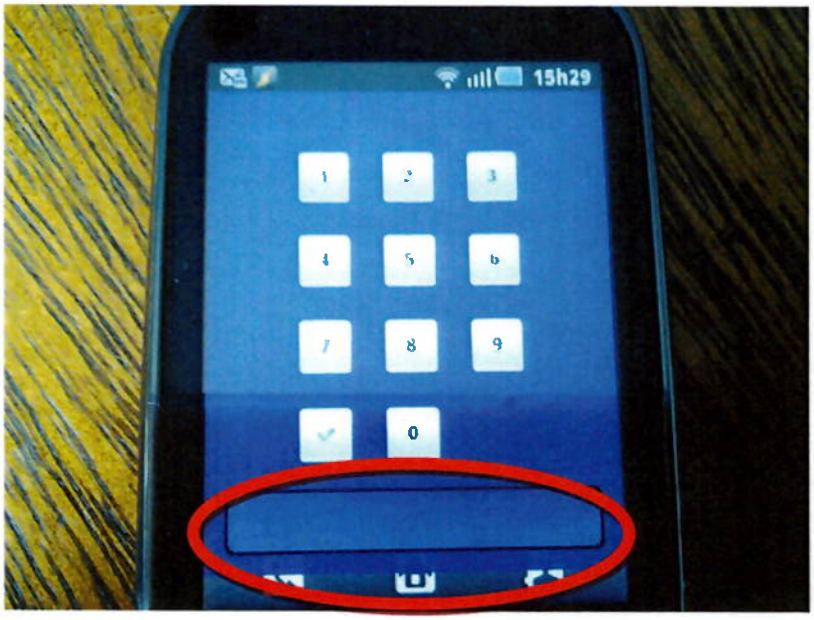


Figura 30. Foto quando o dispositivo se encontra fora da área do local "GPS casa"

Fim do teste: A duração da bateria foi de 17 horas.

Execução do Teste 5

Acrescentam-se mais quatro regras sensíveis ao contexto (Regras para o Teste 5) para as mesmas utilizações propostas no teste 4.

Com 7 horas após o inicio da carga de 100% da bateria (sempre no mesmo horário) uma solicitação 3G, uma ligação de 30 segundos (sempre para o mesmo telefone), receber uma ligação (sempre do mesmo telefone), enviar um SMS (sempre a mesma mensagem e para o mesmo telefone) e ligar o fone de ouvido, reproduzindo 30 segundos de áudio (sempre o mesmo áudio) Executar a mesma utilização descrita acima 7 horas depois da primeira utilização. Entre estas utilizações o dispositivo permaneceu em repouso.

A ativação das regras do teste acima é ilustrada pelas Figura 31.

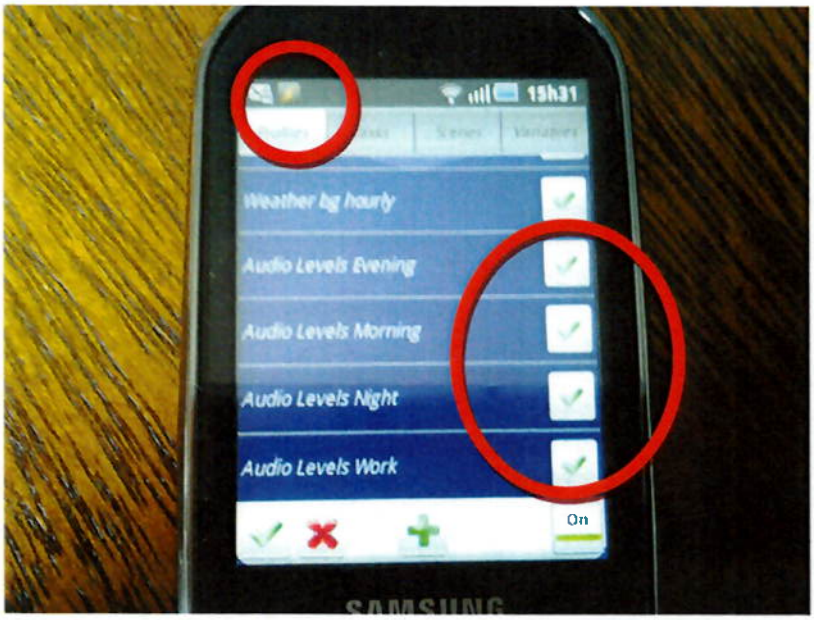


Figura 31. Foto Ativação das quatro e últimas regras para o Teste 5.

4.3 Resultados

Os resultados dos testes realizados podem ser verificados na Figura 32.

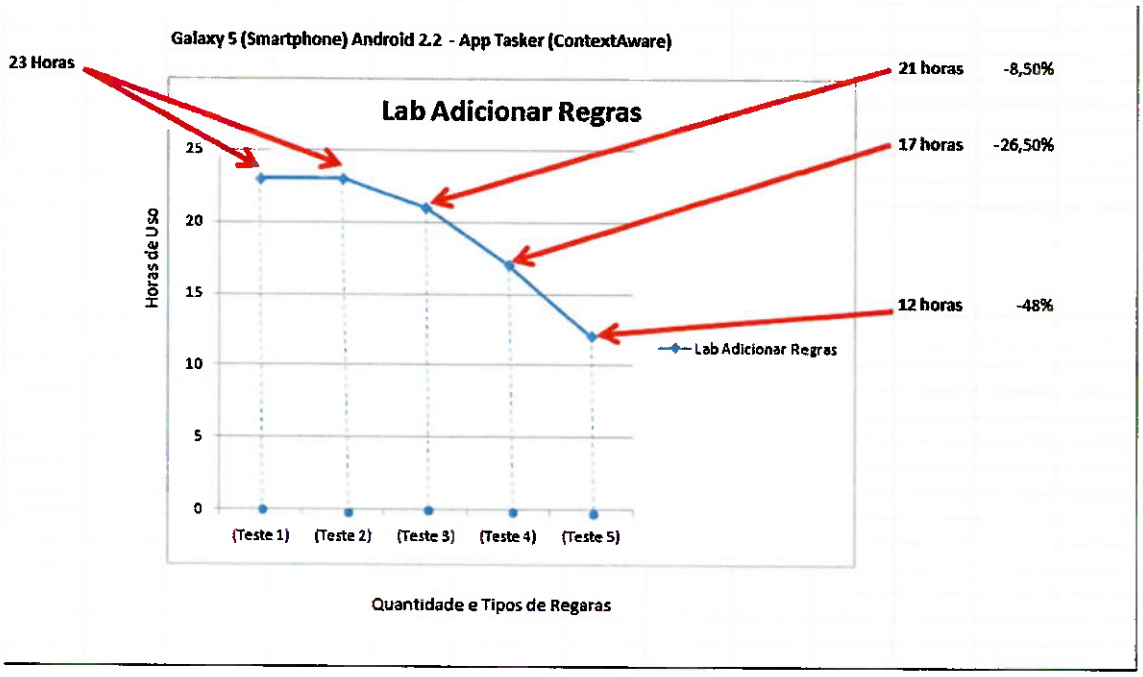


Figura 32. Resultado do Teste com o dispositivo móvel Galaxy 5 (SAMSUNG.GALXY 5, 2012).

Pudemos observar que quando o dispositivo não tem regras sensíveis ao contexto ativas a duração da bateria é de 23 horas com a execução das utilizações propostas.

Quando o dispositivo tem a adição de três regra de contexto (Regras Para o Teste 2) observa-se que a duração bateria se repete em 23 horas com a execução das utilizações propostas.

Ao acrescentarem-se mais três regras de contexto (Regras Para o Teste 3) com a execução das mesmas utilizações propostas, observa-se que a duração da bateria foi de 21 horas. Um consumo aproximadamente 8,5 % maior de energia da bateria.

Acrescentando-se mais uma regra de contexto (Regras Para o Teste 4) para as mesmas utilizações do segundo e terceiro testes, observa-se que a duração da bateria é de 17 horas, uma diminuição de 26,5 %. É importante observar que

esta regra aciona o GPS com uma freqüência de 15 minutos para determinar a localização do dispositivo.

No quinto e último teste acrescentaram-se mais quatro regras sensíveis ao contexto (regras para o Teste 5) para as mesmas utilizações do dispositivo já descritas, observando-se que a duração da bateria é de 12 horas, com diminuição de 48% em relação à não utilização de regras sensíveis ao contexto no dispositivo. Neste teste, só foi possível executar o primeiro ciclo de utilização, pois a energia da bateria do dispositivo terminou antes do tempo de execução do segundo ciclo de utilização.

O consumo de energia não aumenta linearmente enquanto se aumenta o número de regras ativas, mas como já era esperado ao adicionarmos um maior número de regras, o consumo de energia da bateria foi maior. Estes pontos também foram observados em Conti, Nguyen e Crispo (2010). Um maior consumo de energia foi verificado quando a regra ativada utiliza recursos caros em termos de energia, que em função da quantidade de solicitações aos recursos do dispositivo e combinado a um maior numero de regras provoca a diminuição de energia. A utilização de coordenadas GPS ou conexão 3G, por exemplo, provocam um consumo maior de energia da bateria se acionados com freqüência nas regras sensíveis ao contexto.

Em função do descrito e verificado em Conti, Nguyen e Crispo (2010) e dos resultados verificados no experimento proposto pode-se afirmar que o consumo de energia é uma das principais questões a ser solucionada em dispositivos móveis, quando da utilização de regras sensíveis ao contexto.

Verificou-se que otimizações na programação das regras são possíveis particularmente em relação a obtenção do resultado da regra ativada, utilizando recursos com menor consumo de energia. Por exemplo, a verificação da localização por coordenadas GPS utilizada na Regra para o Teste 4 poderia ser otimizada aproveitando-se uma interface Wi-Fi disponível em um roteador na rede local da casa do usuário do dispositivo, sem a utilização do GPS, tendo a identificação da localização acionada pelo sinal Wi-Fi e com isto, um menor consumo de energia da bateria.

Um conjunto de regras sensíveis ao contexto com utilização freqüente dos recursos que consomem mais energia da bateria, GPS e 3G, por exemplo, se implementadas sem levar o consumo de energia em consideração, pode provocar uma experiência de usuário com seu dispositivo insatisfatória por um consumo exagerado desta energia, como também foi verificado em (CONTI, NGUYEN, CRISPO, 2010).

A utilização destas mesmas regras com os cuidados de otimização citados anteriormente podem garantir uma melhor experiência do usuário com um consumo satisfatório de energia mesmo com um conjunto maior de regras sensíveis ao contexto ativas, onde a proteção e o controle de utilização desejados é alcançada.

4.4 Considerações do Capítulo

Foi descrita a pesquisa de referência, o planejamento e execução do experimento proposto (feito de forma isolada e controlada), os cinco testes e medições.

Verificou-se que os resultados obtidos na pesquisa de referência se confirmaram no experimento proposto.

O número de regras sensíveis ao contexto não promovem um aumento no consumo de energia de forma linear e sim, quando da ativação destas regras que acionam recursos do dispositivo móvel caros em termos de consumo de energia, como as coordenadas GPS ou o sinal 3G, por exemplo.

No próximo capítulo será apresentada as contribuições deste trabalho e possíveis trabalhos futuros a partir dos resultados verificados.

5 CONSIDERAÇÕES FINAIS

Com a apresentação dos resultados já descritos anteriormente no trabalho, este Capítulo visa descrever as contribuições do mesmo dentro da proposta e objetivos iniciais.

Também são sugeridos desdobramentos e trabalhos futuros que poderão desenvolver os resultados verificados.

5.1 Contribuições do Trabalho

Nos dias atuais, em que a mobilidade se encontra a cada dia mais presente e importante para as necessidades de usuários e negócios tem-se como desafio prover a utilização destes dispositivos com proteção e controle de sua utilização de maneira adequada, de modo a proporcionar uma experiência do usuário com seu dispositivo de acordo com seu interesse.

Ao aplicar regras sensíveis ao contexto para possibilitar uma utilização do dispositivo móvel mais flexível, com maior proteção e automatizada com o objetivo de obter uma melhor experiência do usuário destacou-se a importância da gestão do consumo de energia da bateria para atender de forma adequada esta utilização.

Assim, o trabalho apresentou que a utilização de um número maior ou menor de regras sensíveis ao contexto não provoca um consumo de energia linear proporcional. Como descrito no Capítulo 4, a ativação de regras sensíveis ao contexto que acionam recursos do dispositivo móvel caros em termos de consumo de energia como as coordenadas GPS ou o sinal 3G, são normalmente os responsáveis pelo aumento deste consumo. Se estes recursos forem gerenciados e mecanismos alternativos forem utilizados pode-se obter uma otimização no consumo de energia e, conseqüentemente, a proteção do dispositivo com uma maior e mais satisfatória experiência para o usuário.

Também foram descritos exemplos e sugestões que sugerem que a partir da comparação dos resultados descritos no capítulo 4 é possível implementar regras sensíveis ao contexto que se utilizam da verificação de localização do

dispositivo aproveitando outras informações por meio do provedor de uma operadora de telefonia ou interfaces Wi-Fi disponíveis (como o roteador de uma rede local) ou, ainda, com o GPS sendo acionado de forma mais esporádica.

Outro tema apresentado foram as técnicas de gestão do consumo de energia dos dispositivos móveis. Estas técnicas são também implementadas utilizando-se as regras sensíveis ao contexto. Tais regras de gerenciamento de energia tem como objetivo especifico proporcionar um controle da utilização da energia da bateria do dispositivo móvel por parte das aplicações e, em alguns casos, a partir do histórico de comportamento deste consumo de energia. De maneira automática, podem inclusive escolher que serviços devem ser desativados.

Na pesquisa de referência e literatura em geral, não foram descritas as técnicas de controle de utilização de dispositivos móveis por regras sensíveis ao contexto, em combinação com outras técnicas de proteção e gestão de energia da bateria em função da utilização das mesmas. Como principal contribuição deste trabalho, descrevemos estas combinações.

Este trabalho apresenta um experimento para verificação da influência dessas técnicas e uma comparação com os resultados obtidos na pesquisa de referência. A partir da análise desta comparação e seus resultados sugere pontos de atenção em relação a utilização dessas técnicas, destacando tal combinação em análise mais ampla para a maximização da experiência do usuário.

Para que a utilização do dispositivo móvel seja adequada, visando promover uma melhor experiência do usuário, em particular, no quesito proteção de informações, no acesso ou na utilização dos recursos disponíveis. Regras sensíveis ao contexto podem ser ativadas observando-se os aspectos de aproveitamento dos recursos e sensores do dispositivo visando um menor consumo de energia da bateria, proteção e controle de utilização.

Em resumo, os conceitos e técnicas discutidos neste trabalho propõem que a utilização de regras sensíveis ao contexto para proteção e controle de

utilização do dispositivo móvel visando uma experiência do usuário satisfatória é possível e adequada.

5.2 Trabalhos Futuros

A ferramenta Tasker utilizada no experimento proposto não é de utilização simples, suas regras sensíveis ao contexto devem ser programadas e exigem um usuário avançado para sua implementação. Apesar deste tipo de ferramenta ser muito flexível e eficiente, seria mais adequado que novas soluções permitissem uma utilização mais simples para usuários comuns.

Estas novas soluções poderiam prover um mecanismo de implementação das regras sensíveis ao contexto que tivessem a capacidade de identificar, implementar, ativar e controlar regras sensíveis ao contexto adequadas para uma melhor experiência de uso do dispositivo, baseada na analise continua do histórico do comportamento de utilização do dispositivo móvel pelo usuário.

Este mecanismo também poderia identificar a melhor gestão dos recursos de energia da bateria do dispositivo para a utilização desejada, podendo esta mesma gestão ser proporcionada por uma ou mais regras sensíveis ao contexto.

Se o dispositivo móvel puder reconhecer, que o software de vídeo está funcionado com interrupções na sua execução, em 15 quadros por segundo e não nos 30 ideais de forma automática, ele pode corrigir este problema. A idéia é construir uma aplicação ou adicionar ao sistema operacional Android a capacidade de detectar quando suas aplicações estão rodando muito devagar e considerar as possíveis soluções.

Se o dispositivo estiver com a bateria completamente carregada, talvez o sistema operacional possa direcionar maior capacidade computacional para o aplicativo. Caso não esteja, o sistema operacional seria capaz de fazer o aplicativo usar um conjunto de instruções de pior qualidade, porém mais eficiente. Além disso, o sistema operacional pode aprender com a experiência,

de modo a solucionar o problema mais rápido na vez seguinte. Um dispositivo autoconsciente seria capaz de distribuir tarefas complexas como "rode estes três programas mas dê prioridade ao primeiro" ou "economize o máximo de energia possível, desde que isso não interfira na qualidade do filme que estou tentando assistir ou áudio que escuto".

O desenvolvimento desta camada de software projetada como uma aplicação adicional ao sistema operacional Android, que funcione ininterruptamente e que possa modelar os recursos usados por qualquer aplicativo poderia ser realizado (por exemplo, através de heurísticas ou lógicas de detecção de padrões de comportamento das regras sensíveis ao contexto). Se o vídeo do dispositivo móvel estiver rodando lentamente, o sistema operacional poderá alocar mais energia para esse aplicativo. Mas, se estiver rodando a 40 quadros por segundo, o dispositivo poderá desviar energia para outra operação, porque a imagem não terá melhor qualidade aos olhos humanos se o vídeo rodar a 40 quadros por segundo em vez de 30 e economizar energia da bateria.

REFERÊNCIAS

ANDROID MARKET. Loja Android Market na web. Disponível em: < https://market.android.com/?hl=pt_BR >. Acesso em: 10/01/2012.

BAI, G.; GU, L.; FENG, T.; Guo, Y. . Contex-Aware Usage Control for Android.IEEE Institute of Software, School of EECS, Peking University. Beijing. China. 2010. p. 1-18.

CANALYS. Empresa de consultoria. Relatório digital .Disponível em: <**http://www.canalys.com/newsroom/google%E2%80%99s-android-becomes-world%E2%80%99s-leading-smart-phone-platform**>. Acesso em: 20/08/2011.

CNET. Empresa de consultoria. Relatório digital .Disponível em: < **http://www.cnet.com/**>. Acesso em: 21/08/2011.

CONTI, M.; NGUYEN, V. T. N.; CRISPO, B. . CREPE Context-Related Policy Enforcement for Android. Netherlands. Vrije Universiteit Amsterdam, NL. 2010. p. 1-15.

DEVELOPER ANDROID. Site oficial para desenvolvedores do Android. Fornece o Android SDK e documentação para desenvolvedores de aplicativos e designers. Disponível em: <**http://developer.android.com/resources/dashboard/platform-versions.html**>. Acesso em: 03/12/2011

ENCK, W.; OCTEAU, D.; MCDANIEL P.; CHAUDHURI, S. . A Study of Android Application Security. Pennsylvania State University. December. 2010. p. 1-16.

GARTNER. Empresa de consultoria. Relatório digital .Disponível em: <**http://www.gartner.com/technology/home.jsp**>. Acesso em: 19/08/2011.

LEAVIT N. . Mobile Security Finally a Serious Problem. 0018-9162/11/ IEEE Computer Society. 2011. p. 11-14.

LI, B.; IM, E. G. . Smartphone promising battlefield for hacker.IEEE Journal of

Security Engineering. Department of Electrorics Computer Engineering, Hanyang University. South Korea. 2011. p. 89-110.

MYLOOKOUT. Empresa de consultoria. Relatório digital .Disponível em: < **https://www.mylookout.com/**>. Acesso em: 21/08/2011.

NORTON. Empresa de soluções de segurança. Relatório digital .Disponível em: < **http://br.norton.com/security_response/malware.jsp**> e <**http://br.norton.com/security_response/phishing.jsp**> Acesso em: 20/08/2011.

RODRIGUEZ, N. V.; CROWCROFT, J. . ErdOS Achieving Energy Savings in Mobile OS .Computer Lab University of Cambridge, United Kingdom. 2011. p. 37-42.

SANSUNG. Empresa de soluções de mobilidade. Documentação técnica .Disponível em: <**http://www.tgdaily.com/**>. Acesso em: 05/01/2012.

SHASBTAI, A.; FLEDEL, Y.; KANONOV, U.; ELOVICI, Y; DOLEV, S. . Google Android: A State of the Art Review of Security Mechanisms.IEEE Department of Computer Science. Ben-Gurion University, Israel. 2010. p. 1-42.

STUECKLE D. J. . Android Protection System: A Signed Code Security Mechanism for Smartphone Applications. Degree of Master of Science in Computer Engineering Air Force Institute of Technology. Wright-Patterson Air Force Base, Ohio USAF. 2011. p.1-98.

TASKER. Site da empresa de solução de aplicação de ativação de regras sensíveis ao contexto. Biblioteca de regras .Disponível em: < http://tasker.dinglisch.net/index.html> e < http://tasker.wikidot.com/profile-index> Acesso em: 01/01/2012.

TGDAILY. Empresa de consultoria. Relatório digital .Disponível em: <**http://www.tgdaily.com/**>. Acesso em: 21/08/2011.

BIBLIOGRAFIA COMPLEMENTAR

WHIPPLE, J.; ARENSMAN, W.; BOLER, M. S. . A Public Safety Application of GPS Enabled Smartphones and the Android Operating System. Information Systems Engineering Department Southwest Research Institute, San Antonio, Texas, USA. 2009. p. 2059-2061. 978-1-4244-2794-9/09/ IEEE.

BUTLER, M. .Android Changing the Mobile Landscape. Intel Labs. Published by the IEEE CS n 1536-1268/11/ 2011 IEEE. 2011. p. 4-7.

ENCK, W.; ONGTANG, M.; MCDANIEL, P. .Understanding Android Security. Pennsylvania State University.USA. Published by the IEEE Computer Society. 2009. p. 50-57. 1540-7993/09/ IEEE.

HU, D. H.; DONG, F.; WANG, C. . A Semantic Context Management Framework on Mobile Device. Department of Computer Science. The University of Hong Kong. 2009. p. 331-338. 978-0-7695-3678-1/09 IEEE.

HU, W.; CHEN,T.; SHI, Q; LOU, X. . Smartphone Software Development Course Design Based on Android. College of Computer Science Zhejiang University Hangzhou, Zhejiang, P.R.China. 2010. p. 2180-2184. 978-0-7695-4108-2/10 IEEE.

JETER, L.; MANI, M.; REINSCHMIDT, T. . Smart Phone Malware: The danger and protective strategies. Department of Compute Science University of Colorado at Boulder.USA. 2010. p.1-11.

KUNDU, T. K.; PAUL, K. . Android on Mobile Devices An Energy Perspective. Dept. of Computer Science & Engineering, IIT Delhi, India. 978-0-7695-4108-2/10 IEEE.p. 2421-2426. 2010.

KUNDU, T. K.; PAUL, K. . Improving Android performance and energy effiency. Dept. of Computer Science & Engineering, IIT Delhi, India. 2011.p. 256-261. 1063-9667/11 IEEE.

LA, H. J.; KIM, S. D. . A Service based Approach to Developing Android Mobile Internet Device (MID) Applications. Department of Computer Science Soongsil University. Sangdo-Dong, Dongjak-Ku, Seoul. Korea. 2009. 978-1-4244-5299-6/09/ IEEE.

ETTINGER, M. V.; LIPTON, J.; NELWAN, S.; DAM, V. T.; PUTTEN, N. V. D. . Multimedia Paging for Clinical Alarms on Mobile Platforms. Computing in Cardiology. Netherlands. 2010. p. 57–60. ISSN 0276–6574.

SACHSE N. R. S. . Avaliação Comparativa do Modelo de Proteção do Android, Dissertação para obtenção do grau de Mestre em Computação Móvel, Universidade Fernando Pessoa. Brasil. 2010.

OBERHEIDE, J.; VEERARAGHAVAN, K; COOKE, E.; FLINN, J.; JAHANIAN, F. .Virtualized InCloud Security Services for Mobile Devices. Electrical Engineering and Computer Science Department. University of Michigan, Ann Arbor, MI 48109. USA. 2009. p. 1-5.

ONGTANG, M.; MCLAUGHLIN, S.; ENCK, W.; MCDANIEL, P. .Semantically Rich Application-Centric Security in Android. Department of Computer Science and Engineering. The Pennsylvania State University, University Park. USA. 2009. p. 340-349. IEEEDOI 10.1109/ACSAC.2009.39.

PROFFITT, B. . Tools&Toys. spectrum.ieee.org. 2011. p. 22-24.

SCHUSTER, D.; SPRINGER, T.; SCHILL, A. . Service-based Development of Mobile Real-time Collaboration Applications for Social Networks. TU Dresden, Computer Networks Group Dresden, Germany. 2010. p. 1-6.

TESHOME, S. F. . Spying Software Development in Google Android. Helsinki Metropolia University of Applied Sciences. Bachelor of Engineering. Degree Programme in Information Technology. Finlândia. 2010. p. 1-53.

TYCHALAS, D. ;KAKAROUNTAS, A. . Planning and Development of an Electronic Health Record Client based on the Android Platform. Dpt. of Computer Science and Biomedical Informatics University of Central Greece

Lamia, Greece. p. 3-6 ,2010. 978-0-7695-4172-3/10 IEEE.

WALLACH, D. . Smartphone Security: Trends and Predictions. Rice University. 2011. USA. p. 1-11.

WU, B. ; WANG, A. I.; RUUD, A. H. . Extending Google Android's Application as an Educational Tool. Norwegian University of Science and Technology, Norway; Guilin University of Electronic Technology, China. 2010. p. 23-30. 978-0-7695-3993-5/10 IEEE.

WU, Y.; LUO, J.; LUO, L. .Porting mobile web application engine to the Android platform. School of Computer Science and Engineering,University of Electronic Science and Technology of China. 2010. p. 2157-2161 978-0-7695-4108-2/10 IEEE.

XIONG, H.; YAO, D. D.; HAN, L.; IFTODE, L. .Personal Anomaly Detection and SmartPhone Security. Department of Computer Science, Virginia Tech Blacksburg VA 24060 and Rutgers University Piscataway NJ 08854. USA. 2010. p. 1-6.

GLOSSÁRIO

Malware - *Malware* é uma categoria de código malicioso que inclui vírus, worms e Cavalos de Tróia. Os programas de malware destrutivos utilizam ferramentas de comunicação conhecidas para se espalharem. O *malware* também tenta explorar as vulnerabilidades existentes nos sistemas, tornando sua entrada discreta e fácil (NORTON,2011).

Phishing - *Phishing* é basicamente um golpe on-line de falsificação, e seus criadores não passam de falsários e ladrões de identidade especializados em tecnologia. Eles usam spams, websites maliciosos, mensagens instantâneas e de e-mail para fazer com que as pessoas revelem informações sigilosas, como números de contas bancárias e de cartões de crédito (NORTON,2011).

Context-Aware Computing – A empresa de consultoria Gartner define computação sensível ao contexto como o conceito de aproveitar as informações sobre o usuário final e melhorar a qualidade da interação. Emergentes serviços enriquecidos com contexto devem utilizar localização, presença, os atributos sociais, e outras informações ambientais para antecipar as necessidades imediatas de um usuário final, oferecendo o mais sofisticado, a situação consciente e funções úteis.

Firewall - Um *firewall* proporciona um meio para que as organizações criem uma camada entre as redes de tal forma que elas fiquem completamente isoladas de redes externas, tal como a *Internet*, e estejam completamente conectadas a outras. Usualmente colocados entre a rede interna e a rede externa de uma organização, o *firewall* provê um meio simples para controlar o tamanho e os tipos de tráfego que irão passar entre as duas redes.

SO Linux - Linux é ao mesmo tempo um kernel (ou núcleo) e o sistema operacional que roda sobre ele, dependendo do contexto em que você encontrar a referência.

O Linux adota a GPL, uma licença de software livre – o que significa, entre outras coisas, que todos os interessados podem usá-lo e redistribuí-lo, nos termos da licença.

O sistema funciona em dezenas de outras plataformas, entre as quais há expoentes como o sistema Android, mantido pelo Google.

Wi-Fi - Comumente o termo Wi-Fi é entendido como uma tecnologia de interconexão entre dispositivos sem fio,usando o protocolo IEEE 802.11. Atualmente praticamente todos os computadores portáteis vêm de fábrica com dispositivos para rede sem fio no padrão Wi-Fi (802.11b, a ou g).

GPS - O Sistema de Posicionamento Global (GPS) é um sistema de navegação baseado em satélite, composto de uma rede de 24 satélites colocada em órbita pelo Departamento Norte-Americano de Defesa.

GPS trabalha em qualquer condição de tempo, em qualquer lugar no mundo, 24 horas por dia, e não é cobrada nenhuma taxa para se usar o GPS.

3G – Padrão para conexão dos celulares de terceira geração. O principal atrativo deste novo padrão é a maior velocidade de transmissão de dados. Estamos falando de 2 megabits, contra apenas 14.4 k do Wap e 144 k dos celulares 2.5G.