UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO PARA PERSONALIZAÇÃO DA PRIVACIDADE EM DISPOSITIVOS MÓVEIS

LEONARDO LEITE DE MELO

ORIENTADOR: Prof. Dr. Sérgio Donizetti Zorzo

São Carlos - SP
Julho/2013

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO PARA PERSONALIZAÇÃO DA PRIVACIDADE EM DISPOSITIVOS MÓVEIS

LEONARDO LEITE DE MELO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes.

Orientador: Prof. Dr. Sérgio Donizetti Zorzo.

São Carlos - SP
Julho /2013

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

M528mp

Melo, Leonardo Leite de.
Mecanismo para personalização da privacidade em dispositivos móveis / Leonardo Leite de Melo. -- São Carlos : UFSCar, 2013.
114 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2013.

1. Privacidade e personalização. 2. Android (Recurso eletrônico). 3. Controle de acesso. 4. Sistemas de segurança. I. Título.

CDD: 004.6 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Mecanismo para a Personalização da Privacidade em Dispositivos Móveis"

Leonardo Leite de Melo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:

Prof. Dr. Sérgio Donizetti Zorzo
(Orientador - DC/UFSCar)

Prof. Dr. Hélio Crestana Guardia
(DC/UFSCar)

Prof. Dr. João José Neto
(POLI/USP)

São Carlos
Setembro/2013

DEDICO ESTE TRABALHO À MINHA FAMÍLIA, MINHA MAIOR INSPIRAÇÃO E MOTIVAÇÃO, MEU ALICERCE E PORTO SEGURO DURANTE TODOS OS MOMENTOS, ONDE SEMPRE PUDE BUSCAR CONFORTO NOS MOMENTOS MAIS DIFÍCEIS.

AGRADECIMENTOS

Agradeço aos meus pais, Maria Teresa e Moacyr, e à minha irmã, Andressa, por sempre estarem ao meu lado, apoiando-me de todas as formas que sempre precisei, incentivando-me e ajudando-me a seguir em frente não só durante todo o curso de mestrado, mas em toda a vida.

Agradeço ao meu orientador, Prof. Sérgio Donizetti Zorzo, pela oportunidade e confiança; por todo auxílio, experiência e conhecimento que me proporcionou durante o período de pesquisa. Durante o curso de mestrado pude, sem dúvida, conhecer melhor o meio de pesquisa acadêmica, aumentar meu conhecimento e conhecer pessoas extraordinárias.

Por fim, agradeço a todos que, de forma direta ou indireta, participaram do meu caminho até a conclusão deste trabalho.

O que eu ouço, eu esqueço. O que eu vejo, eu lembro. O que eu faço, eu entendo.

Confúcio

RESUMO

O avanço tecnológico presente em dispositivos móveis, tais como câmera, sistema de localização por GPS, conexão à Internet, tem permitido a instalação de aplicações que utilizam esses recursos possibilitando o acesso a uma quantidade quase ilimitada de informação. Alguns usuários possuem em seus dispositivos móveis mais dados sensíveis que aqueles contidos em seus computadores pessoais, gerando uma preocupação crescente com a possibilidade de que tais informações possam ser utilizadas de forma maliciosa, resultando em danos por roubo de informações ou vulnerabilidade física. Este trabalho apresenta uma proposta de ferramenta que possibilita a transferência do controle de permissão de acesso para o usuário de dispositivos móveis, bem como visa conscientizá-lo sobre o que cada aplicação está executando. Essa ferramenta pode ser empregada por um usuário individual ou por empresas para o gerenciamento da segurança dos dispositivos de seus funcionários. A implementação do protótipo do mecanismo para personalização da privacidade em dispositivos móveis, utilizando a plataforma Android, bem como as avaliações de uso e desempenho realizadas a fim de validar a proposta são apresentadas nesta dissertação.

Palavras-chave: Android. Controle de Acesso. Segurança. Privacidade e Personalização.

ABSTRACT

Technological advances present in mobile devices feature such as camera, GPS tracking system, internet connection which has enabled the installation of applications that use these resources enabling access to almost unlimited amount of information. Some users have on their mobile devices more sensitive data than those contained on their personal computers, generating a growing concern about the possibility that such information can be used maliciously, resulting in damages for theft of information or physical vulnerability. This dissertation proposes a tool that allows to transfer for the user of mobile devices the control of access permission, as well as aims to make user to be aware of what each application is running. This tool can be used by an individual or by companies to manage security devices of their employees. A prototype implementation of the mechanism for customization of privacy on mobile devices using the Android platform as well as the use and performance assessments conducted in order to validate the proposal are presented in this dissertation.

Keywords: Android. Access Control. Security. Privacy and Personalization.

LISTA DE FIGURAS

Figura 1 – Componentes da arquitetura do Android.....17
Figura 2 – Ciclo de vida de uma Atividade.....19
Figura 3 – Exemplo de estrutura de objetos *view*.....20
Figura 4 – Ciclo de vida do serviço 22
Figura 5 – Exemplo de um ICC 24
Figura 6 – Exemplo de AndroidManifest.xml 25
Figura 7 – Exemplo de permissão para receber SMS 26
Figura 8 – Permissão solicitada no momento da instalação da aplicação Maps.....27
Figura 9 – Exemplo de UID compartilhado.....30
Figura 10 – Arquitetura SAINT 37
Figura 11 – Interface de Usuário MockDroid 40
Figura 12 – Arquitetura UAMDroid 41
Figura 13 – Arquitetura CRePE.....42
Figura 14 – Fluxo sem bloqueio 46
Figura 15 – Fluxo com bloqueio 47
Figura 16 – XML com regra personalizada 49
Figura 17 – Ilustração de uma regra 50
Figura 18 – Arquitetura MPP 53
Figura 19 – Alteração no mecanismo de segurança 55
Figura 20 – Desvio para chamada ao mecanismo 56
Figura 21 – Verificação da permissão pelo mecanismo 57
Figura 22 – Extensão da API de Administração 58
Figura 23 – Extensão da classe de administração do Android.....59
Figura 24 – Método Add criado para adicionar uma nova regra.....59
Figura 25 – Tela inicial (a) e Tela de personalização (b).....62
Figura 26 – Alerta de bloqueio 62
Figura 27 – Modelo de Segurança iOS 65
Figura 28 – Etapas do avaliação de uso 67
Figura 29 – Comparativo entre as repostas apresentadas na Tabela 3.....77

LISTA DE TABELAS

Tabela 1 – Dispersão das respostas e Média Ponderada para o Questionário 1 73
Tabela 2 – Dispersão das respostas e Média Ponderada para o Questionário 2 74
Tabela 3 – Respostas para as perguntas presentes nos dois questionários 76
Tabela 4 – Tempo consumido para bloquear o acesso à Internet no Browser 85
Tabela 5 – Tempo consumido para liberar acesso à Internet no Browser 85
Tabela 6 – Tempo consumido para bloquear envio SMS.....86

LISTA DE QUADROS

Quadro 1 - Políticas suportadas pela API de administração 29
Quadro 2 – Permissões de acesso 48
Quadro 3 – Exemplo de variáveis nas condições.....49
Quadro 4 – Quadro de significado da respostas para a escala de sete pontos 70
Quadro 5 – Significância estatística 79
Quadro 6 – Teste T - questão 1 80
Quadro 7 – Teste T – questão 2.....81
Quadro 8 – Teste T – questão 3.....81
Quadro 9 – Teste T - questão 4 82
Quadro 10 – Teste T - questão 5 83

LISTA DE ABREVIATURAS E SIGLAS

.dex	Dalvik Executable
API	Application Programming Interface
HP	Hewlett-Packard
HTC	High Tech Computer Corporation
I/O	Input/output
ICC	Inter-Component Communication
MPP	Mecanismo de Personalização da Privacidade
SDK	Software Development Kit
SO	Sistema Operacional
SQL	Structured Query Language
XML	Extend Markup Language

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO 11

1.1 Considerações iniciais e motivação 11
1.2 Objetivo 13
1.3 Estrutura da dissertação 14

CAPÍTULO 2 - ANDROID 16

2.1 Plataforma Android 16
2.2 Componentes Android 18
2.3 Ativação de componentes 23
2.4 Manifesto 24
2.5 Android: Segurança e Permissões 25
2.6 API de administração do dispositivo 29
2.7 Utilização do framework Android 30
2.8 Considerações Finais 31

CAPÍTULO 3 - MODELOS DE SEGURANÇA 32

3.1 Princípios de Saltzer e Schroeder 32
3.2 Princípios de Dennis e Van Horn 34
3.3 Considerações Finais 35

CAPÍTULO 4 - TRABALHOS RELACIONADOS 36

4.1 SAINT 36
4.2 Apex 38
4.3 TraintDroid 39
4.4 MockDroid 39
4.5 UAMDroid 40
4.6 CRePE 42
4.7 Considerações finais 43

CAPÍTULO 5 - MECANISMO DE PERSONALIZAÇÃO DA PRIVACIDADE 44

5.1 Mecanismo 44
5.2 Arquitetura 47
5.3 Implementação 51
5.4 Protótipo 60

5.5 Considerações Finais 63

CAPÍTULO 6 - EXPERIMENTOS E RESULTADOS 66

6.1 Avaliação de Uso 66
6.1.1 Metodologia 67
6.1.2 Avaliação dos resultados 71
6.1.3 Considerações finais 83
6.2 Avaliação de desempenho 84

CAPÍTULO 7 - CONCLUSÕES E TRABALHOS FUTUROS 87

REFERÊNCIAS 90

APÊNDICE I - CENÁRIOS DE USO 95

APÊNDICE II - APRESENTAÇÃO 101

APÊNDICE III - QUESTIONÁRIO 1 102

APÊNDICE IV - QUESTIONÁRIO 2 105

APÊNDICE IV - SLIDES DE APRESENTAÇÃO DO MECANISMO 112

Capítulo 1

CAPÍTULO 1 - **INTRODUÇÃO**

Este capítulo apresenta a introdução ao trabalho desenvolvido durante o curso de mestrado em Ciência da Computação, listando as considerações iniciais, motivação, objetivo e, por fim, a organização desta dissertação.

1.1 **Considerações iniciais e motivação**

Nos últimos anos, é possível notar a atualização de dispositivos móveis para uma gama cada vez mais diversificada de propósitos, como o uso pessoal, educacional e até mesmo comercial. Essa expansão pode estar associada à diversidade de recursos incorporados nesse tipo de dispositivo (GPS, câmera, *internet* e etc.) e o preço acessível praticado pelos fabricantes.

As aplicações desenvolvidas para esses dispositivos exigem a concessão de permissões, como o acesso à *Internet*, a dados privados do usuário, à conta de e-mail, às redes sociais e, até mesmo, a contas bancárias. Devido à característica de fácil utilização, essas aplicações têm atraído um número crescente de usuários.

Tais facilidades e funcionalidades presentes nesses dispositivos fazem com que muitos usuários tenham mais informações pessoais (fotos, emails, SMSs etc.) nesses dispositivos que em seus computadores pessoais.

Há um contínuo e forte crescimento no número de dispositivos móveis conectados à Internet móvel (dispositivos pessoais e aplicações máquina-a-máquina): em 2017 excederá o número previsto de 7,6 bilhões de pessoas na Terra – segundo a Organização das Nações Unidas (ONU).

De acordo com dados publicados pela cisco, o trafego de dados móveis deve crescer 300% até 2017. Como qualquer outro dispositivo com acesso à *Internet*, os dispositivos móveis fornecem riscos de exposição de informações do usuário, como acontece com os computadores.

Em sua maioria, esses dispositivos possuem recursos de *software* e *hardware* que possibilitam não só o roubo de informações sensíveis, mas também o fornecimento da posição geográfica definida através do GPS, a gravação de ligações telefônicas, a captura de imagem ou da lista de contatos, podendo, inclusive, comprometer a integridade física de seu usuário se esses dados forem acessados por aplicações mal intencionadas. Por exemplo, a posição de GPS pode ser rastreada para determinar a rotina de uma pessoa e utilizada para realizar um assalto ou sequestro.

Dados estes fatos, a motivação para este trabalho é apresentar a proposta de um mecanismo para a personalização da privacidade por meio da transferência do controle de permissões de acesso para o usuário, introduzindo o conceito de regras de acesso personalizadas, que é a habilidade de uma pessoa definir, de acordo com suas necessidades, como os recursos de seu dispositivo são utilizados pelas aplicações.

Assim, busca-se disponibilizar mais privacidade e segurança no acesso realizado por aplicações a recursos e dados contidos em dispositivos móveis.

Privacidade é definida no dicionário com a intimidade de uma pessoa, porém, no contexto computacional, essa definição pode ser extrapolada para o conceito de que a privacidade é a habilidade de uma pessoa de controlar a exposição e a disponibilidade de informações acerca de si.

De acordo com Hughes (1993), privacidade é o poder de revelar-se seletivamente ao mundo. De modo semelhante, Kuhlen e Rainer (2004) dizem que a privacidade não significa apenas o direito de ser deixado em paz, mas também o direito de determinar quais atributos de si serão usados por outros.

A motivação para o desenvolvimento da proposta aqui apresentada é fundamentada no fato de que a configuração de acesso concedido às aplicações, nos sistemas operacionais presentes em dispositivos móveis, em geral é realizada de forma genérica, pois as permissões solicitadas pela aplicação devem ser concedidas pelo usuário no momento da instalação e não podem ser aceitas de

forma parcial, bem como não é fornecido um mecanismo para o gerenciamento das permissões de acesso concedidas às aplicações após instaladas no dispositivo.

Tendo em vista a diminuição da vulnerabilidade da segurança e o aumento da privacidade dos proprietários, busca-se um mecanismo que possibilite uma maior restrição ao acesso de tais informações por terceiros, por meio da transferência de controle de permissões ao usuário, disponibilizando a possibilidade de realizar o gerenciamento e a personalização das permissões de cada aplicação.

O mecanismo proposto e testado tem o intuito de ser genérico, podendo ser implementado para todas as plataformas atualmente executadas em dispositivos móveis.

Mecanismo, neste trabalho, é a proposta de ferramenta, e o protótipo é a implementação física do mecanismo. Prova de conceito pode ser interpretada como a prova de que a proposta funciona.

O protótipo implementado como prova de conceito do mecanismo utilizou a plataforma Android, por esta ser de código livre e de fácil customização, possibilitando alteração em todas as camadas de seu framework.

Outro motivo que contribuiu para a escolha da plataforma na implementação do protótipo é o crescente número de usuários e a diversidade de dispositivos utilizando a plataforma.

Além das funcionalidades citadas, outra aplicabilidade que motiva a proposta aqui apresentada é a possibilidade de que equipes de suporte de corporações utilizem a proposta para desenvolver uma aplicação para o gerenciamento da segurança dos dispositivos de seus funcionários.

1.2 **Objetivo**

O objetivo deste trabalho é apresentar uma proposta de arquitetura e a implementação de um mecanismo de transferência do controle de acesso para o usuário de dispositivos móveis, por meio de regras personalizadas, promovendo, assim, garantias de privacidade e segurança.

O protótipo do mecanismo proposto, desenvolvido como prova de conceito, faz a extensão do modelo existente de segurança da plataforma Android e visa

garantir que as informações e o hardware dos dispositivos não sejam utilizados sem o consentimento de seu proprietário.

Esse mecanismo irá realizar a mediação e a verificação de acesso a dados e hardware em tempo de execução, baseado no que foi personalizado pelo usuário através da interface gráfica fornecida, que é a forma pela qual o usuário poderá criar suas personalizações de acesso de cada aplicação.

Baseado no que foi personalizado pelo usuário, o mecanismo irá liberar ou bloquear o acesso. No momento em que um bloqueio ocorrer, o mecanismo apresenta uma mensagem informativa. A principal funcionalidade das mensagens é informar o usuário sobre o bloqueio; caso deseje, ele pode usar a interface e desbloquear.

No caso de aplicações corporativas, o desenvolvedor poderá utilizar aplicações que se conectem a repositórios de regras da empresa e automaticamente atualizar as regras nos dispositivos. A maior vantagem da utilização desse mecanismo por empresas é a possibilidade da configuração de regras alteradas de forma não intrusiva a seus usuários, garantindo, assim, que todos os dispositivos utilizados em suas redes estejam em conformidade com as políticas da empresa.

A avaliação do protótipo desse mecanismo baseia-se em testes para aferir o tempo consumido para realizar a verificação de acesso.

A avaliação de facilidade de uso do protótipo implementado baseia-se na análise de participantes que utilizaram o mecanismo em cenários de testes e de respostas a dois questionários.

Busca-se, com essas avaliações, mensurar se o protótipo do mecanismo proposto tem atraso na realização das operações em dispositivos móveis, se é um atraso aceitável e se tem usabilidade considerada fácil pelos usuários.

1.3 **Estrutura da dissertação**

Esta dissertação está organizada da seguinte forma: O Capítulo 2 apresenta a plataforma Android com suas principais características e aspectos de segurança. No Capítulo 3 são mostrados os modelos que, de acordo com o estudo realizado, vêm sendo utilizados como referência para implementação de mecanismos de segurança

para dispositivos móveis e outras plataformas com controle de acesso. O Capítulo 4 apresenta os trabalhos relacionados, descrevendo suas principais funcionalidades. O Capítulo 5 apresenta o mecanismo desenvolvido, descrevendo suas características, arquitetura, projeto de implementação e os benefícios que pode proporcionar a seus usuários. E, por fim, o Capítulo 6 apresenta os experimentos e resultados obtidos, bem como propostas para trabalhos futuros e a consideração final.

Capítulo 2

CAPÍTULO 2 - **ANDROID**

Este capítulo descreve os principais conceitos relacionados à Plataforma Android, que foi utilizada para a implementação da prova de conceito e para o ambiente da avaliação de uso realizadas com o mecanismo apresentado nesta dissertação de mestrado.

2.1 **Plataforma Android**

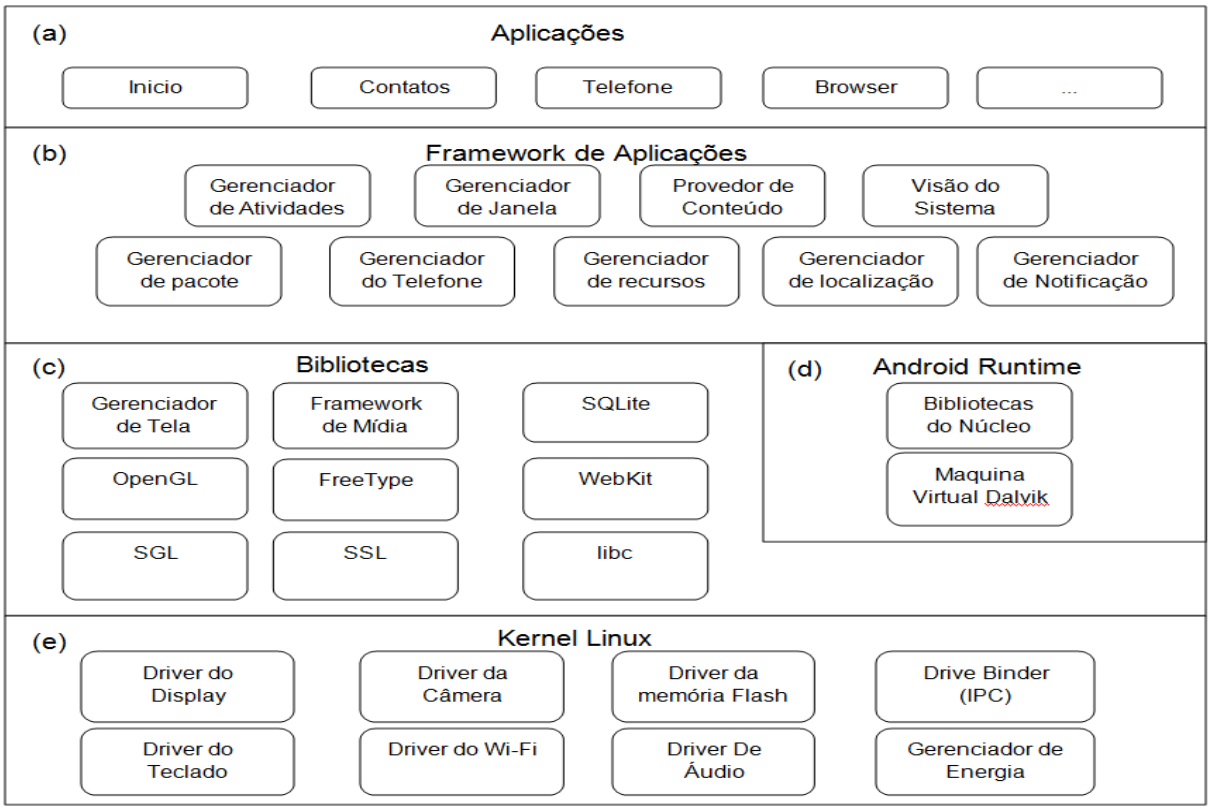
Android é uma pilha de software para dispositivos móveis que inclui um sistema operacional, *middleware* e aplicações chave, especificado pela *Open Hadset Aliance* e liderado pelo Google, em parceria com outros grandes fabricantes de dispositivos móveis, como Samsung, HP, HTC, Sony, Dell, Intel, Motorola, entre outros. O Android SDK(Software Development Kit - Kit de desenvolvimento de software) fornece ferramentas e uma API (sigla de Application Programming Interface - interface de programação de aplicativo) para o desenvolvimento de aplicações para a plataforma Android, utilizando a linguagem de programação Java.

O núcleo do Android é um kernel Linux, apresentado na camada (e) da Figura 1, cuja principal função é gerenciar *drivers* de *hardware*, a memória e interfaces de redes. Em raros casos, os programadores trabalham na camada do kernel.

Acima do Kernel, o Android possui um conjunto de bibliotecas ilustradas na camada (c) da Figura 1. Essas bibliotecas são escritas em C/C++ e são utilizadas por diversos componentes na camada de aplicação. Como exemplo, podemos citar a habilitação da câmera ou do Wi-Fi. A camada de biblioteca nativa contém classes C/C++ customizadas, incluindo uma solução de SQL (Structured Query Language),

bibliotecas 2D e 3D, o browser nativo e codecs de mídia. A principal funcionalidade dessa camada é a disponibilização de recursos para a camada apresentada a seguir.

Figura 1 – Componentes da arquitetura do Android



Fonte: ANDROID, 2012a.

Acima da camada de bibliotecas nativas estão o ambiente de tempo de execução da máquina virtual Dalvik e as bibliotecas do núcleo, ilustradas na camada (d) da Figura 1. A máquina virtual Dalvik é responsável por executar as aplicações Android e aceita uma extensão especial de arquivos, .dex (Executáveis Dalvik), que são mais compactos e eficientes no uso de memória que classes Java. Essa é uma característica relevante para as aplicações, pois a maioria dos dispositivos possuem pouca capacidade de bateria e memória.

As classes das bibliotecas do núcleo do Android, que disponibilizam funções para a construção de novos aplicativos, são escritas em Java. O Android também disponibiliza uma grande quantidade de recursos presentes no Java 50 SE (ex: *Collection*, funções de I/O, rede e utilitários). Essas classes estão ilustradas na

camada (d) na Figura 1, assim como algumas funções específicas do sistema Android estão representadas pela máquina virtual Dalvik. A comunicação entre as aplicações executadas e o sistema operacional é realizada através das bibliotecas nativas e controladas pelo Android.

Na camada (b) da Figura 1 está o framework Android, uma plataforma aberta para o desenvolvimento de novas aplicações. Devido a essa característica, o Android possibilita aos desenvolvedores criarem aplicações utilizando e estendendo componentes já existentes. Desenvolvedores estão livres para utilizar todo o potencial do hardware e do sistema operacional ao criar novas aplicações.

Desenvolvedores têm acesso às mesmas bibliotecas utilizadas pelas aplicações disponibilizadas com o Android, sendo que a arquitetura das aplicações existentes, como pesquisa na Internet e envio de e-mail, foi desenvolvida para facilitar o reuso de componentes na criação de novas aplicações.

O apresentado em (a) da Figura 1 é a camada de aplicações. Android é disponibilizado com algumas aplicações do núcleo, como cliente de e-mail, SMS, calendário, mapas, entre outras. A maioria das novas aplicações é desenvolvida nessa camada.

Aplicações Android são um conjunto de componentes empacotados em arquivos ".apk", muito semelhante a um arquivo ".jar" da linguagem Java.

Cada aplicação executa em seu próprio processo Linux, permanecendo, assim, os dados da aplicação isolados das demais aplicações executadas, impossibilitando acessos não autorizados. As aplicações só podem se comunicar através dos recursos fornecidos pela API do Android. Cada aplicação desenvolvida para o Android deve listar todas as permissões requeridas em seu arquivo AndroidManifest.xml. Esse arquivo contém várias informações sobre a aplicação, incluindo as permissões requeridas, e é utilizado no momento da instalação para apresentar e requisitar a autorização pelo usuário.

2.2 **Componentes Android**

O framework de aplicações do Android permite que se criem aplicativos extremamente ricos em relação à sua apresentação e inovadores, a partir da

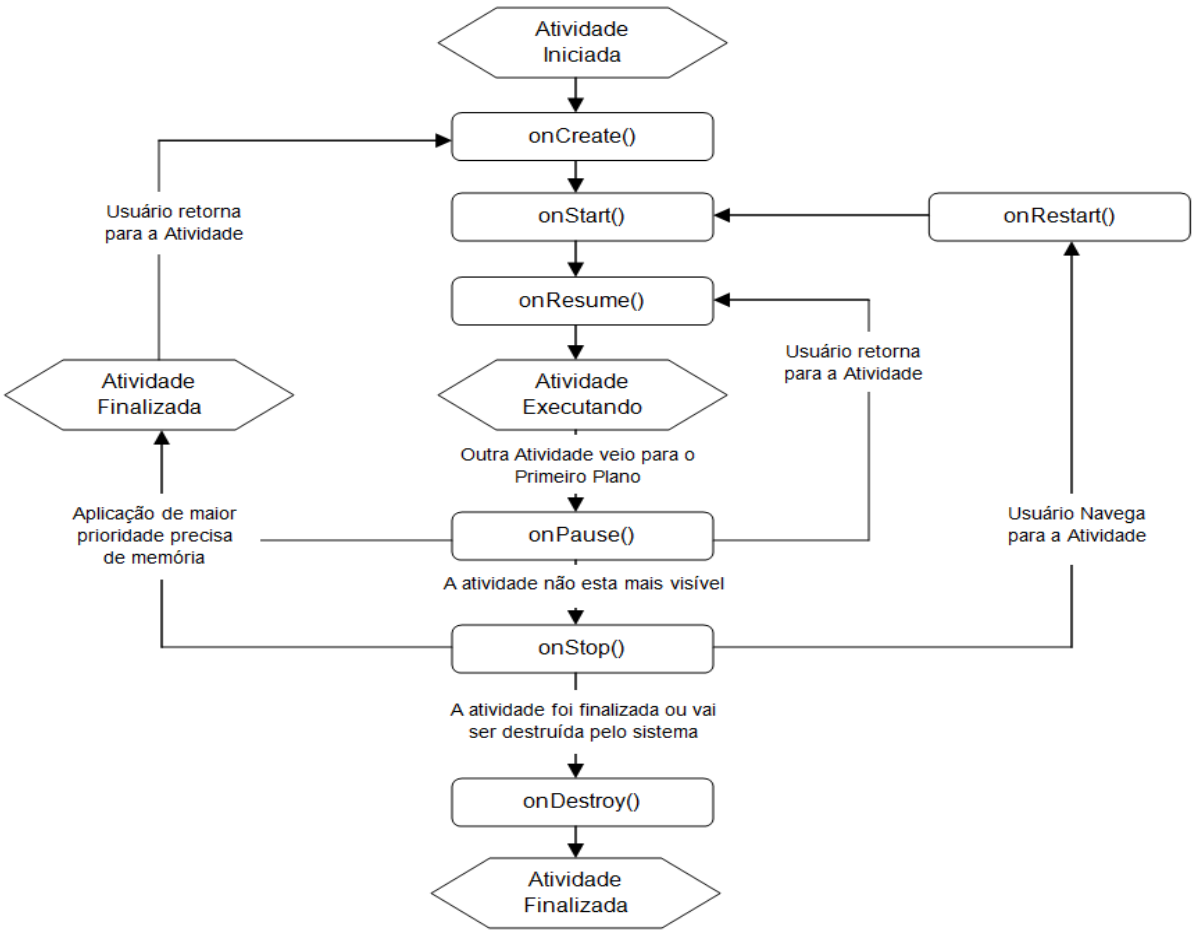
composição e extensão de componentes existentes ou criação de novos componentes. Como exemplo de reutilização pode-se citar a busca na web, que é um componente e pode ser utilizado por outras aplicações.

O Android fornece, basicamente, quatro tipos de componentes para a criação de novos componentes e aplicações, cada um deles desenvolvido com funcionalidades e capacidades específicas. Os componentes básicos fornecidos pelo Android são: Atividade, Serviço, Difusor e Provedor de dados, que são detalhados a seguir.

Atividade

A atividade é um componente que fornece uma tela onde a interface é definida, que é a forma pela qual o usuário pode interagir com a aplicação para realizar uma ação, como mandar um e-mail, fazer uma ligação ou tirar uma foto.

Figura 2 – Ciclo de vida de uma Atividade



Fonte: ANDROID, 2012a.

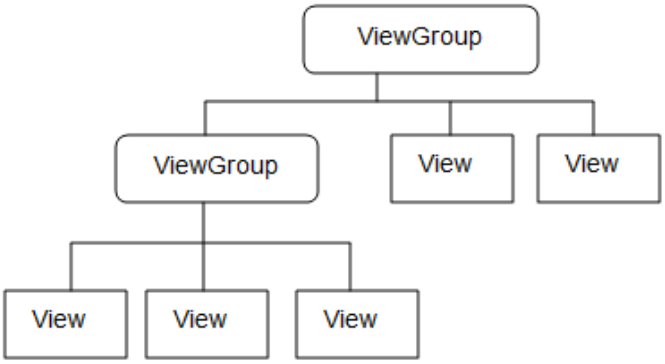
Tipicamente, uma aplicação é composta por diversas atividades. Define-se na aplicação uma a ser executada quando é iniciada e, a partir dessa "atividade inicial", pode-se executar a transição para as demais que compõem a aplicação (ANDROID, 2012a).

Para criar uma nova atividade, deve-se estender a classe *Activity*, disponibilizada pelo framework, e, na subclasse, devem-se implementar os métodos de *callback* que o framework executa de acordo com a mudança de estados no ciclo de vida, como apresentado na Figura 2.

A Figura 2 apresenta todos os possíveis estados que uma atividade pode assumir durante o período em que está ativa. A atividade assumirá um determinado estado dependendo das ações que a aplicação está executando. Em cada um dos estados apresentados é possível adicionar código para gerenciar os dados da aplicação ou realizar uma ação.

A definição da interface pelo usuário é feita através de uma estrutura hierárquica de objetos do tipo *View*, que derivam da classe *ViewGroup*, como apresentado na Figura 3. A plataforma fornece um grande número de objetos desse tipo, que podem ser utilizados na construção de novas interfaces.

Figura 3 – Exemplo de estrutura de objetos *view*



Fonte: ANDROID, 2012e.

O modo mais comum de definir a interface é por meio de um arquivo XML de *layout*, salvo com os recursos da aplicação. Desse modo, a interface fica separada do código da atividade. A Figura 3 apresenta um exemplo de uma composição de objetos do tipo *view* para a construção de uma interface. Nesse exemplo temos um

objeto principal do tipo *ViewGroup* utilizado para agrupar objetos do tipo *View* para construir uma interface.

Serviço

Serviço é um componente que executa, em segundo plano*,* geralmente processamentos de longa duração ou tarefas de processos remotos, como um *download*. Serviços não fornecem interface ao usuário (ANDROID, 2012) e são iniciados por uma Atividade ou Evento.

Serviços são lançados por outros componentes e continuam executando em segundo plano até finalizar sua tarefa, mesmo que o usuário inicie a utilização de outra aplicação. Um serviço em execução pode aceitar associação por outros componentes e executar ações. Exemplos de serviços são *downloads* ou um reprodutor de música executando em segundo plano.

Serviços assumem basicamente duas formas durante seu ciclo de vida:

***Started*:** um serviço assume esse estado quando um componente de uma aplicação (como uma *Activity*) o inicia, chamando o método *startService*. Pode continuar executando até terminar sua tarefa, mesmo que o componente que o criou seja destruído. Os possíveis estados de um serviço quando S*tarted* são apresentados do lado esquerdo da Figura 4.

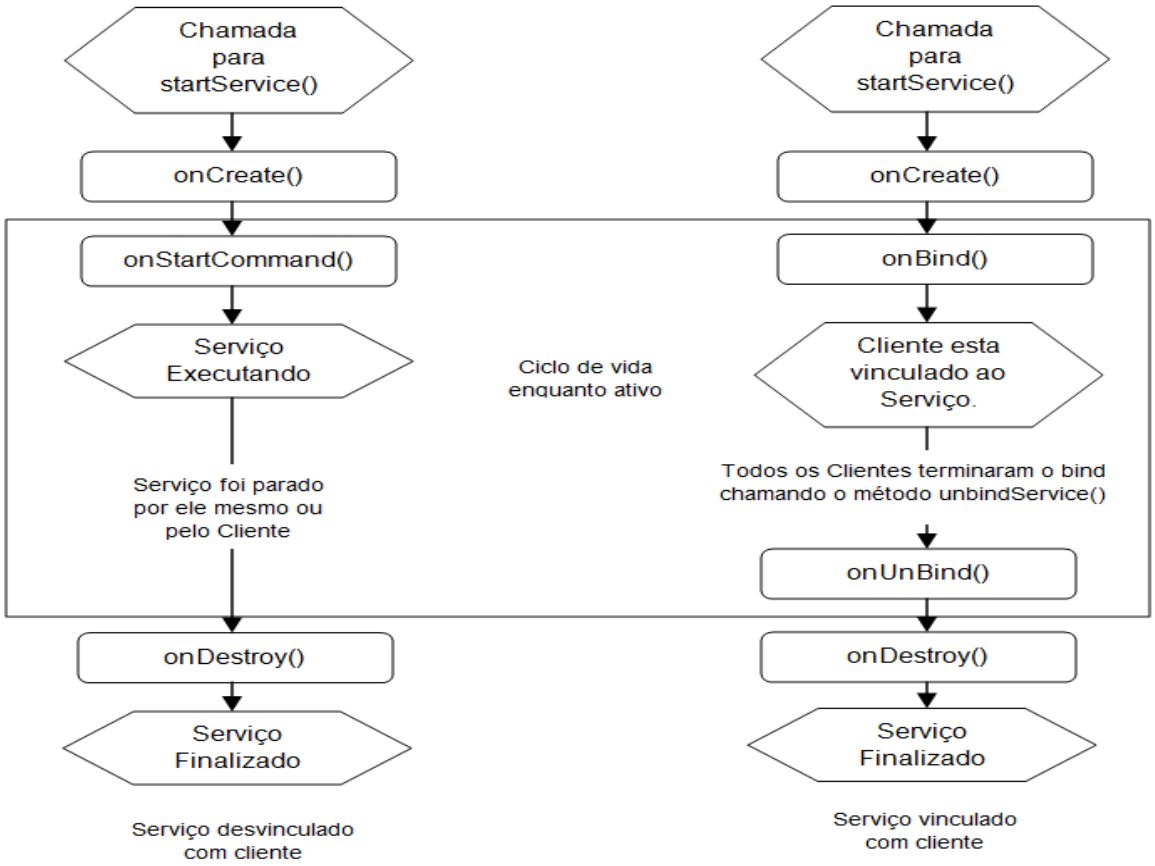
***Bound*:** esse estado é assumido quando um componente faz uma associação com o serviço através do método *bindService*. Serviços em *bound* oferecem uma interface para os componentes interagirem com ele, executando enquanto houver um *bind* ativo com um componente. Os possíveis estados de um serviço quando *Bound* são apresentados do lado direito da Figura 4.

Os serviços podem assumir as duas formas ao mesmo tempo, só dependendo do modo como a classe associada ao serviço for implementada.

A Figura 4 apresenta os métodos de *callback* para cada um dos estados possíveis durante o ciclo de vida de um serviço. Os métodos de *callback* são o modo pelo qual os serviços podem ser invocados para realizar uma determinada tarefa.

Um exemplo seria o método de OnBind executado quando uma nova associação ao serviço é realizada. Nesse método, informações como a aplicação que está solicitando o *Bind* e a ação que deve ser executada poderão ser coletadas.

Figura 4 – Ciclo de vida do serviço



Fonte: ANDROID, 2012a.

Difusor

Na plataforma Android, um difusor geralmente fica esperando por uma notificação e executa uma determinada ação dependendo da mensagem recebida, como o alerta de um novo e-mail recebido (ANDROID, 2012b), uma mensagem de texto ou memória baixa.

Muitas mensagens recebidas pelo difusor são enviadas pelo sistema, como o aviso de que a tela foi desligada. Porém, aplicações também podem iniciar um Difusor, como o aviso do término de um *download*.

Difusores não têm uma interface de usuário e usam a barra de status do sistema operacional para apresentar a mensagem de alerta. Geralmente, Difusores são utilizados como saída para outros componentes e executam uma pequena quantidade de trabalho.

Provedor de dados

Provedores de dados são a interface padrão utilizada pelas aplicações para acessar repositórios de dados (ANDROID, 2012b).Um provedor de dados é responsável por gerenciar o acesso a um conjunto de dados estruturados, encapsulando os dados e provendo mecanismos de segurança. Uma aplicação pode declarar seu próprio provedor de dados ou utilizar o provedor de dados disponibilizado por outras aplicações. Para se comunicar com um provedor de dados é necessário um objeto *ContentResolver* na aplicação. Não é necessária a implementação de provedor de dados próprio em uma aplicação se não se pretende compartilhar os dados com outras aplicações. Porém, será necessária caso se desejem implementar buscas personalizadas ou copiar e colar dados entre aplicações.

2.3 **Ativação de componentes**

Três dos quatro tipos de componentes disponíveis (Atividades, Serviços e Difusores) são ativados por mensagens assíncronas, chamadas de *Intent*, utilizadas para associar dois componentes em tempo de execução.

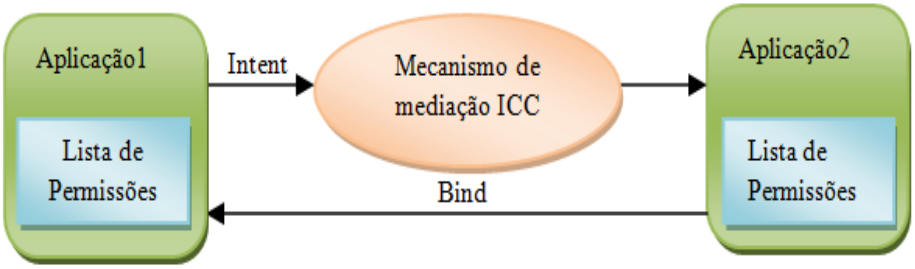
Para a atividade e os serviços, uma *Intent* define uma ação a ser executada. Para os Difusores*,* uma *Intent* é apenas um anúncio sendo enviado (ANDROID, 2012b). *Intents* são criadas como um objeto da classe *Intent* e definem uma mensagem para ativar um componente específico, ou um tipo de componente, como um *browser.* No caso de existir mais de um programa instalado que possa responder à requisição, o usuário deve escolher qual programa quer executar.

ContentProviders não são ativados com *Intent,* mas sim por um *ContentResolver*, que será responsável por receber e executar todas as requisições para o *Content Provider.*

Android utiliza um mecanismo denominado *inter-component communication* (ICC) como um de seus mecanismos de garantia de segurança e, por sua vez, é o foco do mecanismo de segurança proposto neste trabalho. O ICC faz a verificação de segurança dentro do framework Android, habilitando a execução de ações únicas e com permissões explicitamente concedidas.

O mecanismo central responsável pela garantia da segurança em sistemas Android é também responsável por intermediar todos os ICC, como mostrado na Figura 5, que ilustra a interceptação realizada pelo mecanismo de ICC durante uma solicitação de acesso.

Figura 5 – Exemplo de um ICC



Fonte: Elaborado pelo autor.

2.4 **Manifesto**

Um manifesto em sistema Android é um arquivo XML (*eXtensible Markup Language*), intitulado *AndroidManifest.xml,* como exemplificado na Figura 6, e deve estar localizado obrigatoriamente no diretório raiz de uma aplicação. Nesse arquivo estão todas as informações de que o Android necessita para executar uma aplicação, incluindo as permissões solicitadas.

Antes de um componente de uma aplicação ser executado, o Android precisa saber que ele existe. Essas informações do componente são obtidas por meio desse arquivo. O arquivo de manifesto tem diversas funções, além de declarar os componentes de uma aplicação:

a) Identificar as permissões requeridas pela aplicação, como acesso à Internet ou ler a lista de contatos;
b) Declarar o nível requerido da API do Android para executar a aplicação;
c) Declarar os recursos de hardware que a aplicação utiliza, como câmera, wi-fi, dentre outros e
d) Programas dos quais a aplicação depende, como o Google Maps.

Figura 6 – Exemplo de AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
 <application android:icon="@drawable/app_icon.png" ... >
 <activity android:name="com.example.project.ExampleActivity"
 android:label="@string/example_label" ... >
 </activity>
 ...
 </application>
</manifest>
```

Fonte: Elaborado pelo autor.

A Figura 6 apresenta um exemplo simples de um arquivo de manifesto, contendo apenas o ícone da aplicação, a atividade inicial e o título da atividade.

2.5 **Android: Segurança e Permissões**

Android é um sistema de privilégio separado, como no Unix, em que cada aplicação recebe seu próprio UID (User ID) e grupo, que são utilizados para criar o processo no qual a aplicação irá executar e proteger o acesso a variáveis em memória.

Um mecanismo para segurança dos acessos realizados pelas aplicações pode ser fornecido por meio de um sistema de permissões, o qual aplica restrições nas operações que uma aplicação pode executar (ANDROID, 2012d).

O modelo de segurança existente em sistemas Android é descrito a seguir, apresentando a arquitetura do modelo de segurança, o conceito de SandBox, questões de Permissões e níveis de Permissão, assinatura de aplicação, ID de usuário e acesso a arquivos.

Arquitetura do Modelo de Segurança

A arquitetura de segurança do Android define que, por padrão, uma aplicação não tem permissão para executar nenhuma operação que possa impactar alguma outra aplicação no sistema operacional. Por exemplo, acessar a Internet, escrever e ler na lista de contatos e desativar o dispositivo, entre outras (ANDROID, 2012d). Essa característica obriga as aplicações a declararem todas as permissões necessárias para executar, devendo o usuário conceder a permissão no momento da instalação.

O conceito de *SandBox* (ANDROID, 2012d) é utilizado para a execução de uma aplicação. Como todas as aplicações são executadas em seu próprio processo, quando uma nova é iniciada, um processo é concedido a esta, mantendo, assim, a segurança dos dados em tempo de execução.

A plataforma Android, por padrão, executa cada processo em uma *SandBox*. Para entender o conceito de uma S*andBox*, é necessário imaginar uma "caixa de areia" onde pode ser usado tudo que está dentro mas não se pode usar nada a seu redor, protegendo, assim, o ambiente que envolve a caixa.

Para que se possa acessar o ambiente externo, deve ser declarada uma permissão no arquivo (*AndroidManifest.xml*) da aplicação. Exemplos de acesso ao meio externo à *SandBox* que devem ser declarados para que a aplicação seja capaz de acessa-los são a *Internet* e a lista de contatos no dispositivo.

Um fato importante é que o usuário não pode aceitar parcialmente as permissões requeridas pela aplicação, estas devem ser aceitas ou negadas como um todo. A plataforma Android não possui um mecanismo de gerenciamento das permissões instaladas, pois, na especificação, considerou-se que seria mais complicado para o usuário (ANDROID, 2012f).

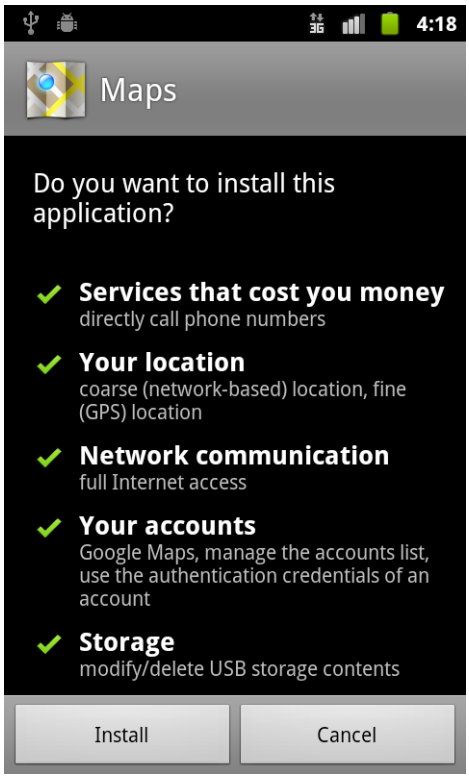
Figura 7 – Exemplo de permissão para receber SMS

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
                                                    package="com.android.app.myapp" >
  <uses-permission android:name="android.permission.RECEIVE_SMS" />
  ...
</manifest>
```

Fonte: ANDROID, 2012f.

A Figura 7 apresenta a declaração de que a aplicação necessita de acesso à funcionalidade de envio de SMS. Esse é o exemplo da solicitação de uma permissão dentro do arquivo de manifesto. A Figura 8 mostra como as permissões solicitadas por uma aplicação são apresentadas no momento de sua instalação.

Figura 8 – Permissão solicitada no momento da instalação da aplicação Maps



Fonte: ANDROID, 2012f.

Devido ao grande número de permissões e à possibilidade de desenvolvedores criarem novas permissões, foram criados Níveis de Permissões, classificadas conforme risco que apresentem:

a) **Normais:** permissões no nível de aplicação que não envolve muito risco, como um jogo que acessa o diretório raiz da aplicação ou necessita criar um novo arquivo no dispositivo;
b) **Perigosas:** permissões de alto risco, que envolvem acesso a dados privados ou funcionalidades que podem alterar o sistema. Um programa que requer essas permissões deve ser autorizado no momento da instalação;

c) **Assinatura:** essa permissão permite acessar informações restritas, como bases de dados ou dados em memória de aplicações que possuem a mesma assinatura e

d) **Assinatura ou Sistema:** um tipo de permissão por assinatura que permite acesso a pacotes instalados no sistema operacional.

O Android fornece um mecanismo de proteção de aplicações por Assinatura de Aplicação. Todas as aplicações Android (.apk) devem ser associadas a um certificado contendo a chave pública associada à chave privada usada para assinar a aplicação, de propriedade do desenvolvedor. Esse certificado identifica o autor da aplicação. Os certificados não precisam ser reconhecidos por uma autoridade, são comuns certificados autoassinados.

O propósito das assinaturas em aplicações Android é distinguir autores de aplicações, o que auxilia o mecanismo de segurança a conceder ou não acesso quando são requeridas autorizações somente com a mesma assinatura (ANDROID, 2012f).

No momento da instalação da aplicação, esta recebe do Android seu ID de Usuario (UID; Linux User ID) único. O UID continua com a aplicação enquanto esta estiver instalada no dispositivo. Em dispositivos distintos, uma aplicação pode ter UID diferente, ou seja, cada aplicação tem um UID diferente em cada dispositivo em que é instalada.

O UID é utilizado no momento da execução para proteger os dados da aplicação. Como a segurança acontece na camada onde cada aplicação possui seu próprio processo e área de memória associados a um UID, códigos de aplicações diferentes não podem normalmente executar no mesmo processo, pois possuem UID diferentes.

Para executar no mesmo processo deve ser usado o atributo "sharedUserId" no *AndroidManifest.xml* das aplicações que irão compartilhar o UID. Fazendo isso, do ponto de vista da segurança, as aplicações serão tratadas como sendo a mesma, com o mesmo UID e acesso a arquivos. Por motivos de segurança, o mesmo UID só é fornecido a duas aplicações ao mesmo tempo (ANDROID, 2012f).

2.6 **API de administração do dispositivo**

A partir da versão 2.1 do sistema Android, foi introduzido o suporte para aplicações corporativas com a inserção de uma nova API para administração do dispositivo. Essa nova API fornece funcionalidades administrativas e permite criar aplicações de segurança que são úteis no meio corporativo, nas quais, profissionais da área de tecnologia da informação (TI) precisam de um controle avançado sobre os dispositivos dos usuários (ANDROID, 2012c).

Quadro 1 - Políticas suportadas pela API de administração

Política	Descrição
Senha Habilitada	O dispositivo solicita pelo PIN ou senha.
Tamanho mínimo da senha	Define o tamanho mínimo para o número de caracteres de uma senha.
Senha alfanumérica requerida	Requer que a senha seja uma combinação de letras e números.
Senha complexa	Requer que a senha tenha letras, números e símbolos especiais.
Número mínimo de letras	A senha deve ter um número mínimo de letras.
Número mínimo de letras minúsculas	Especifica o número mínimo de letras minúsculas que a senha deve conter.
Número mínimo de caracteres não alfabéticos requeridos	O número mínimo de caracteres não alfabéticos que a senha deve conter.
Número mínimo de caracteres numéricos na senha	O número mínimo de caracteres numéricos que a senha deve conter.
Número mínimo de símbolos na senha	O número mínimo de símbolos especiais que a senha deve conter.
Número mínimo de letras em caixa alta	O número mínimo de caracteres alfabéticos em caixa alta.
Tempo de expiração da senha	Especifica em quanto tempo a senha irá expirar, definido em milissegundos.
Restrição de senha de histórico	Proíbe o usuário de repetir as últimas senhas anteriormente configuradas no dispositivo.
Número máximo de senhas erradas	Estabelece o número de vezes que o usuário pode tentar uma senha antes do dispositivo bloquear.
Tempo máximo antes de bloquear	Tempo que o dispositivo deve ficar sem ser usado antes de bloquear a tela.
Armazenamento encriptado requerido	A área de armazenamento deve ser encriptada, se o dispositivo suportar.
Desativar câmera	Especifica que a câmera deve ser desativada.

Fonte: ANDOIRD, 2012c.

Um exemplo de utilização dessa API seria a criação de uma aplicação que controla o tipo de senha que pode ser definida para destravar a tela do dispositivo ou uma aplicação de e-mail.

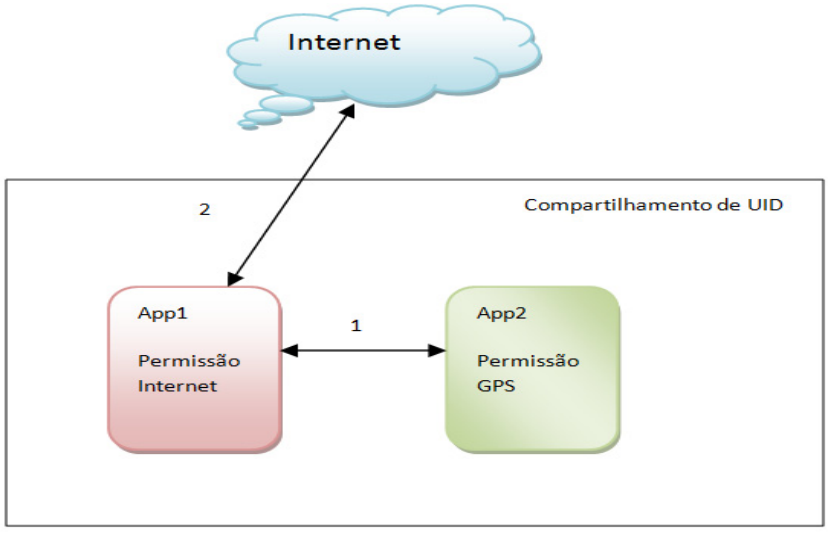
Hoje essa API faz um gerenciamento restrito de segurança, em que a maioria das políticas definidas estão relacionadas com a senha para destravar o dispositivo. A lista completa das políticas de segurança suportadas pela API é apresentada no Quadro 1.

Apesar de as funcionalidades disponíveis atualmente serem básicas, são um bom indicativo de que a plataforma pretende disponibilizar meios para controlar a segurança e, provavelmente, novas funcionalidades vão ser disponibilizadas.

2.7 **Utilização do framework Android**

Com o lançamento do framework Android; devido à sua característica de fácil customização e por ser de código aberto, um número crescente de desenvolvedores vêm criando uma grande variedade de aplicações, que variam de jogos a aplicativos pessoais.

Figura 9 – Exemplo de UID compartilhado



Fonte: Elaborado pelo autor.

Em sua maioria, as aplicações requerem permissões que podem por em risco a segurança de proprietários de dispositivos móveis, como acesso a GPS, *Internet* e lista de contatos, podendo ocorrer casos em que aplicações com código malicioso estejam disponibilizando informações pessoais a terceiros.

Como exemplo desse tipo de problema, podemos citar a situação em que duas aplicações compartilham o mesmo UID App1 e App2 – ilustrada na Figura 9. A aplicação App1 requisita acesso à Internet e a App2, acesso à posição do GPS. Como as duas aplicações irão compartilhar o mesmo UID, o código malicioso pode estar usando a permissão da App2 (seta número 1) para ler a posição do GPS e depois enviar pela *Internet* com a aplicação App1 (seta número 2) como na Figura 9.

2.8 **Considerações Finais**

Os modelo de gerenciamento de permissões da plataforma Android disponibiliza um mecanismo que oferece segurança a seus usuários. Porém, os usuários não estão habilitados a gerenciar a forma com que as aplicações instaladas no dispositivo utilizam suas permissões.

Capítulo 3

CAPÍTULO 3 - **MODELOS DE SEGURANÇA**

Este capítulo apresenta os princípios de segurança utilizados como referência nos mecanismos de segurança e gerenciamento de permissões existentes em sistemas computacionais que embasaram esta proposta.

3.1 **Princípios de Saltzer e Schroeder**

Segundo (SALTZER et al. 1975), um modelo de segurança deve seguir oito princípios básicos de modo a proteger um sistema e as informações de seus usuários. Esses princípios são descritos a seguir:

a) Economia de mecanismos

Esse princípio diz que um mecanismo de segurança deve ser mantido o mais simples e o menor possível com relação à sua codificação, para que seja fácil de ser avaliado e compreendido, evitando assim, erros de codificação que podem comprometer a segurança.

b) Acesso bloqueado por padrão

O sistema deve negar o acesso a recursos, a menos que tenha sido concedido explicitamente, ou seja, nenhum componente ou funcionalidade pode ser executado a menos que uma permissão tenha sido concedida. Um exemplo seria uma aplicação que deve ter o acesso à *Internet* concedido no momento da instalação ou da utilização do recurso.

c) Mediação completa

Todas as solicitações de acesso devem ser verificadas, incluindo as de acesso a recursos pelo SO, forçando assim uma verificação de permissão em todo o sistema.

d) Projeto aberto

A segurança de um sistema não depende de manter seu mecanismo secreto. Sistemas abertos baseiam-se em chaves de encriptação e senhas, não na ocultação da arquitetura. O fato de a codificação de um mecanismo ser de código aberto não implica que proporcione menor segurança.

e) Separação de privilégios

Um mecanismo que utiliza duas chaves é melhor que um mecanismo que utiliza apenas uma. O fundamento é que, tendo cada aplicação uma chave, estas podem ser separadas, sendo assim, organizações e indivíduos podem ser responsabilizados separadamente por elas.

f) Menor privilégio

Uma aplicação deve executar com o menor número de permissões possíveis, de modo a reduzir acessos não autorizados. Esse princípio diz que apenas as permissões realmente necessárias devem ser concedidas à aplicação, evitando, assim, que permissões não necessárias sejam utilizadas para fins maliciosos.

g) Menor número de mecanismos comuns

As aplicações devem utilizar o menor número de recursos compartilhados, de modo a prevenir o roubo de informações. Um mecanismo comum representa um grande fluxo de informação e deve ser desenvolvido com cuidado para evitar o comprometimento da segurança.

Um exemplo seria uma base de dados de mapas compartilhada por diversas aplicações. A implementação de uma base de dados para cada aplicação, mesmo que com informações replicadas, pode proporcionar uma maior segurança.

h) Aceitação psicológica

Esse princípio diz que, para um mecanismo de segurança ser realmente válido, deve ser aceito pelo usuário, pois, em sua maioria, esses mecanismos requerem configuração. Caso a interface com o usuário não seja de fácil utilização, o mecanismo, apesar de eficaz, pode não executar sua funcionalidade, pois não é utilizado da forma correta.

3.2 **Princípios de Dennis e Van Horn**

Segundo Dennis et al. (1966), deve ser criado um modelo de segurança de referência para a proteção que se deseja disponibilizar. É com base nesse modelo que deve ser desenvolvido o mecanismo de segurança pretendido.

Esse estudo também introduz o conceito de que todos os componentes envolvidos (software ou hardware) devem ser capazes de determinar quais recursos protegidos podem ser acessados e qual o risco do acesso. A utilização dos princípios apresentados por Dennis et al. (1966) cria um elo forte na troca de informações entre os componentes, já que o mecanismo está validando os acessos.

Os princípios básicos de um modelo de referência, segundo os autores, são descritos a seguir:

a) Mecanismo deve ser inviolável

Para que um certificado de segurança seja concedido garantindo a eficiência do mecanismo, este deve estar protegido de alterações manuais ou por código de programação de computador. Esse princípio impossibilita que as verificações de segurança realizadas pelo mecanismo sejam corrompidas por alguma alteração manual, como configuração do dispositivo ou por códigos maliciosos.

b) Todos os acessos devem ser verificados

Todas as solicitações de acesso devem ser verificadas pelo mecanismo que realiza a validação de segurança. Este deve ser invocado e aplicado para validar todos as solicitações, incluindo as solicitações provenientes do sistema operacional, inserindo, assim, uma verificação completa a todos os acessos realizados.

c) Mecanismo deve ser pequeno o suficiente para ser sujeito à análise para ser avaliado

A implementação de um mecanismo com a menor quantidade de programação auxilia na sua a verificação. Mecanismos pequenos são mais fáceis de provar que são fiéis a seu modelo, corretamente implementados e, consequentemente, são mais fáceis de serem validados.

3.3 **Considerações Finais**

Um fato a se observar é que, apesar de as publicações que apresentam os princípios serem antigas, estas nos remetem à diferença entre um conteúdo científico defasado e um conteúdo clássico.

Como apresentado em Smith (2012), esses princípios ainda são referência em sistemas de segurança e utilizados em material didático para ensinar os conceitos fundamentais sobre segurança em sistemas computacionais. Pode se notar a criação de novos métodos, porém, todos de alguma forma seguem esses princípios.

Dados esses fatos, com o intuito de desenvolver um mecanismo que possa fornecer maior segurança a usuários de dispositivos móveis, a proposta apresentada neste trabalho seguiu os princípios apresentados neste capítulo.

Entre os motivos para a utilização desses princípios, deve-se ressaltar o fato de serem amplamente conhecidos e serem seguidos pela maioria dos modelos de segurança desenvolvidos para sistemas computacionais, incluindo os instalados em dispositivos móveis.

Como exemplo, podemos citar a plataforma Symbian, que utiliza os modelos como referência. Para as plataformas Android e iOS, não foram encontradas referências aos modelos adotados, mas vários desses princípios foram observados na plataforma Android. A mesma avaliação não pode ser realizada na plataforma iOS, pois é de código fechado.

Capítulo 4

CAPÍTULO 4 - **TRABALHOS RELACIONADOS**

Neste capítulo serão apresentados alguns trabalhos relacionados com o mecanismo apresentado neste trabalho e que subsidiaram esta proposta, bem como a diferença entre os trabalhos existentes e o aqui proposto.

4.1 **SAINT**

Em SAINT (ONGTANG et al, 2009), é apresentada uma semântica detalhada para a definição de regras de segurança baseada em aplicações e estados. A proposta é um *framework* que estende a arquitetura de segurança do Android, incorporando requisitos de segurança constituídos por regras e variáveis de controle.

Na proposta apresentada em SAINT, as aplicações em tempo de instalação proveem políticas que regulam o acesso. Essas políticas definem os acessos requeridos e concedidos por essa aplicação, ou seja, em tempo de instalação a aplicação define se outras aplicações podem acessar seus componentes. Em tempo de execução, a verificação de permissão está sujeita a políticas de segurança de ambos os componentes, o que requer e o requerido.

SAINT vai além da validação de permissões estáticas, possibilitando a restrição baseada no estado corrente do dispositivo como, por exemplo, rede e quantidade de bateria.

Uma aplicação *A* declarando uma permissão *P* de acesso a um de seus componentes define também as condições sobre as quais *P* é concedida para outras aplicações durante a instalação. Isso implica que uma aplicação *B,* que está sendo instalada, requisitando a permissão *P* declarada pela aplicação *A,* será instalada somente se as condições impostas pela aplicação *A,* que declara a permissão, forem satisfeitas.

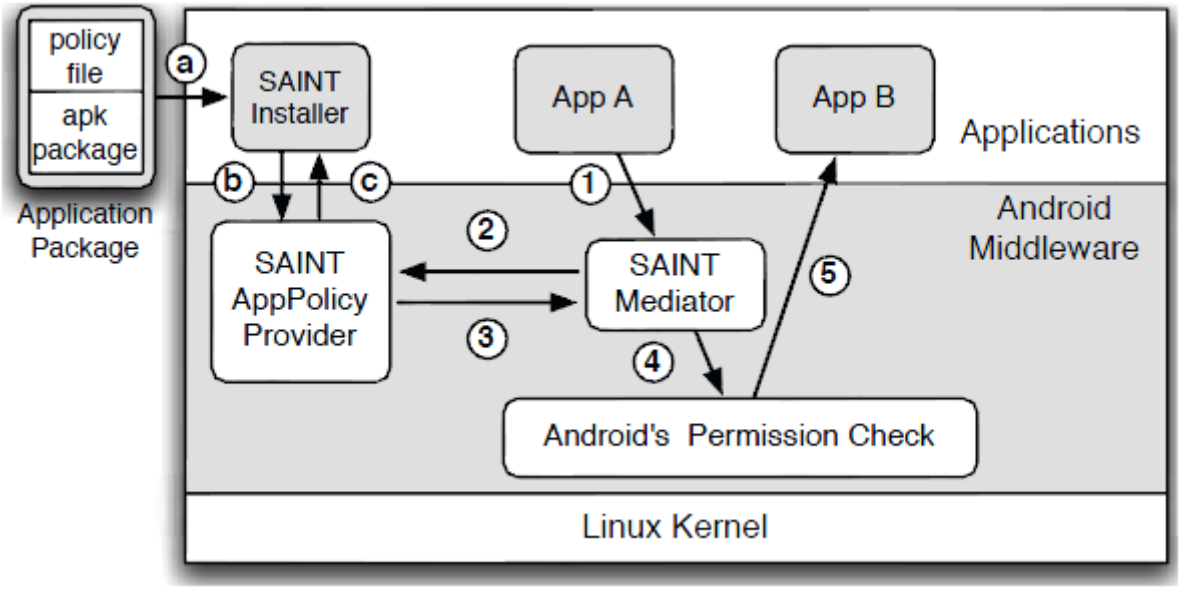
O modelo de permissões do Android concede ou não a permissão requerida, como *Internet*, SMS etc., baseado em regras independentes. SAINT possibilita exercer um controle não só sobre as permissões requeridas, mas também sobre a concessão de permissões.

As políticas de *runtime* definidas por SAINT regulam, em tempo de execução, a interação entre os componentes das aplicações e o *middleware* do Android. Todas as interações entre uma aplicação, que faz uma requisição enviando uma *ICC (Inter Component Communication),* e uma aplicação requisitada, que recebe a *ICC*, só serão permitidas se as políticas especificadas pelos dois lados forem satisfeitas.

A Figura 10 apresenta a arquitetura desenvolvida em SAINT. No momento da instalação, na letra (a), uma nova aplicação que define as permissões requeridas e as concedidas está sendo instalada.

Nas letras (b) e (c), "SAINT *Installer*" verifica as permissões e armazena em seu repositório; a aplicação só é instalada se as permissões forem concedidas. Em tempo de execução, quando uma nova aplicação solicita um acesso – número (1), a requisição é mediada pelo "SAINT *Mediator*" e verificada no repositório de regras "SAINT *AppPolicy Provider*". Caso aprovada, é enviada para o mecanismo do Android - número (4), e em seguida encaminhada ao que foi solicitado - número (5).

Figura 10 – Arquitetura SAINT



Fonte: ONGTANG et al, 2009, p. 344.

4.2 **Apex**

Em Apex (NAUMAN; KHAN; ZHANG, 2010) é apresentada uma proposta para a extensão do modelo de permissões da plataforma Android, centrando-se no usuário através da inserção de restrições definidas e avaliadas em tempo de execução.

Apex propõe, através da inclusão do conceito de regras, uma maior garantia da segurança dos usuários, estabelecendo que as regras de acessos a componentes serão definidas na forma de predicados.

A solução é baseada na introdução de conceitos para o gerenciamento das regras dinamicamente, denominados “atributos de aplicação”. Cada aplicação é associada a um conjunto de variáveis e esse conjunto de variáveis e seus valores ajudam a definir o “estado de uma aplicação”. O estado de cada aplicação é uma estrutura persistente em que os valores de cada “atributo de aplicação” são armazenados.

Apex permite que a definição de regras seja descrita em forma de predicados, que mapeiam o estado atual do dispositivo ao estado de uma aplicação definido por seus "atributos de aplicação", para realizar a verificação de permissão. Tais predicados são tratados na forma de "exclusivos", ou seja, um valor verdadeiro só é retornado se todos os atributos envolvidos no predicado forem avaliados como verdadeiros. A mudança no estado de uma aplicação ocorre a cada vez que os valores de um atributo são atualizados.

Em Apex foram também introduzidas políticas que definem condições sobre as quais uma aplicação tem a permissão de executar o que foi solicitado. Uma política é aplicada a um estado da aplicação e os valores dos atributos em cada estado determinam a veracidade dos predicados, se são satisfeitos ou não.

Sejam as permissões concedidas ou não, a atualização das variáveis de controle irá ocorrer, habilitando, assim, a natureza dinâmica das políticas de permissão.

4.3 **TraintDroid**

TraintDroid (DELAC; SILIC; KROLO, 2011) propõe uma ferramenta para análise das informações contidas em cada fluxo de mensagem de um sistema Android, habilitando, assim, o processamento de dados específicos pela máquina virtual Java, como localização do GPS.

Apesar de o trabalho em questão ser uma boa fonte de referência para os riscos e possibilidades de roubo de informação, foi apresentado um modelo de análise do fluxo de dados e não uma solução para a proteção das informações.

4.4 **MockDroid**

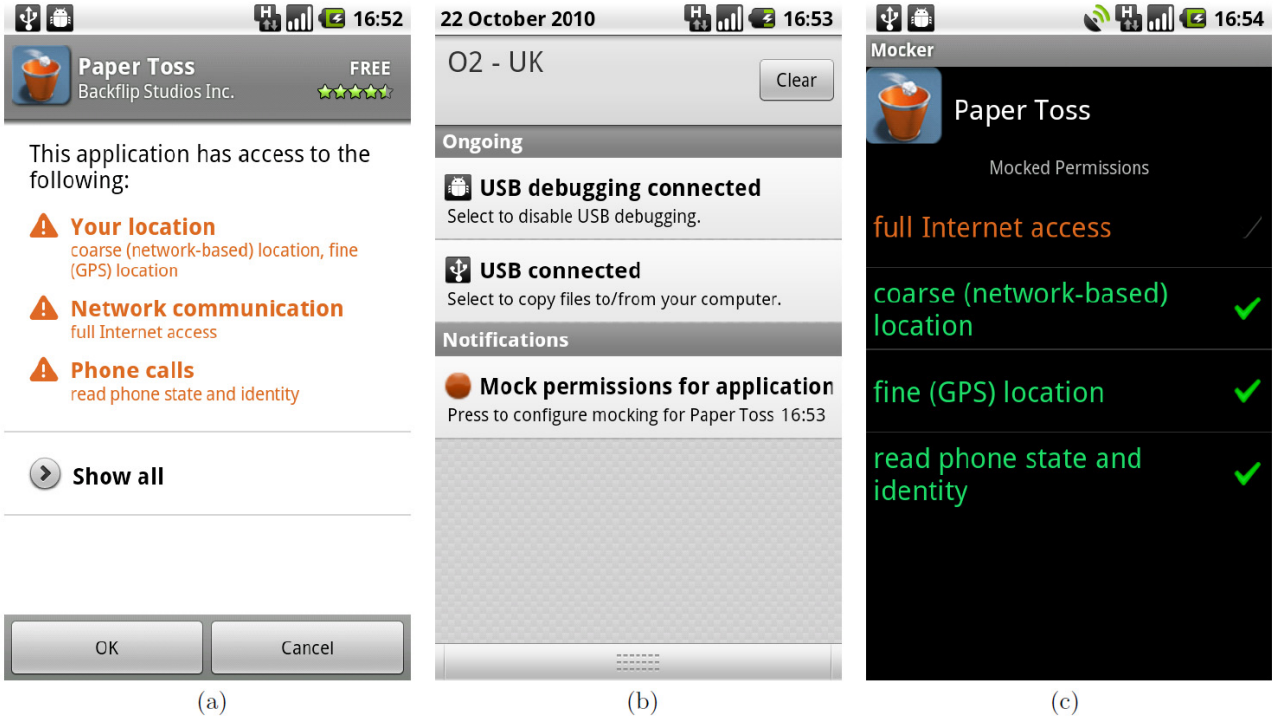
MockDroid (BERESFORD et al, 2012) propõe uma solução de controle da privacidade dos proprietários de dispositivos através da possibilidade de fornecer informações não reais em alguns componentes como, por exemplo, posições falsas para a localização do GPS ou até mesmo o endereço de IP, visando à garantia da privacidade do usuário.

A

Figura 11 apresenta as telas implementadas para o MockDroid, a tela (a) mostra as permissões solicitadas pela aplicação, a tela (b) apresenta a interface para configurar valores falsos para uma aplicação e a tela (c) apresenta os valores falsos configurados para a aplicação.

Um ponto de possível vulnerabilidade, observada neste trabalho, é que, apesar de o autor alegar que, por se tratar de informações falsas, estas não causam danos ao usuário; protocolos como o TCP/IP possuem alguns dados que podem ajudar na localização de um dispositivo, comprometendo a segurança caso a aplicação utilize esse protocolo para enviar as informações falsas.

Figura 11 – Interface de Usuário MockDroid



Fonte: BERESFORD et al, 2012, p. 52.

4.5 **UAMDroid**

Em UAMDroid (LIU; NAM; SHIN, 2011) é apresentada uma solução que permite ao proprietário do dispositivo especificar quais aplicações podem ser acessadas, removidas ou instaladas por um usuário. A solução estende o modelo existente no Android através da inserção de classes de usuários como Administrador, Usuário normal e Convidado. Essas classes de usuários são utilizadas para definir os acessos administrativos sobre o dispositivo (executar, instalar ou remover uma aplicação).

A solução não altera o modelo existente, mas é construída sobre ele. Assim, o sistema operacional não precisa ser recarregado quando é feita a mudança de usuário com relação aos acessos concedidos, permitindo que o proprietário proíba

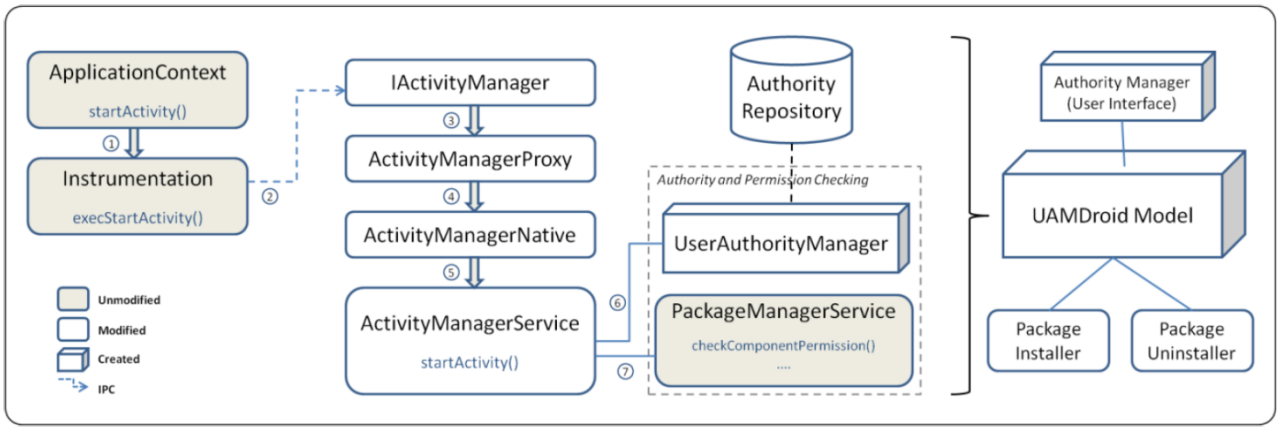
que outras pessoas instalem, desinstalem ou executem aplicações em tempo de execução.

A solução foi implementada por meio da inserção de novos componentes, bem como da alteração de alguns componentes atuais para o gerenciamento da execução de aplicações. UAMDroid permite que aplicações marcadas como protegidas só sejam executadas, removidas ou instaladas se o proprietário tiver acessado o aparelho como Administrador. Uma interface para o proprietário acessar nesse modo é disponibilizada junto com a solução, utilizada para realizar a alteração do usuário utilizando o dispositivo no momento.

Também foram alterados os programas de instalação e remoção de aplicações na plataforma Android para realizar tal controle. Quando se tenta instalar ou remover uma aplicação, é verificado se tal ação é protegida e, em caso afirmativo, o processo não se completa.

O UAMDroid verifica se o usuário tem a permissão de executar antes de enviar para o modelo do Android. Caso a permissão de execução não seja validada pelo UAMDroid, a solicitação é forçada a terminar sem ser enviada para o modelo base. A Figura 12 apresenta as alterações realizadas para a criação do UAMDroid.

Figura 12 – Arquitetura UAMDroid

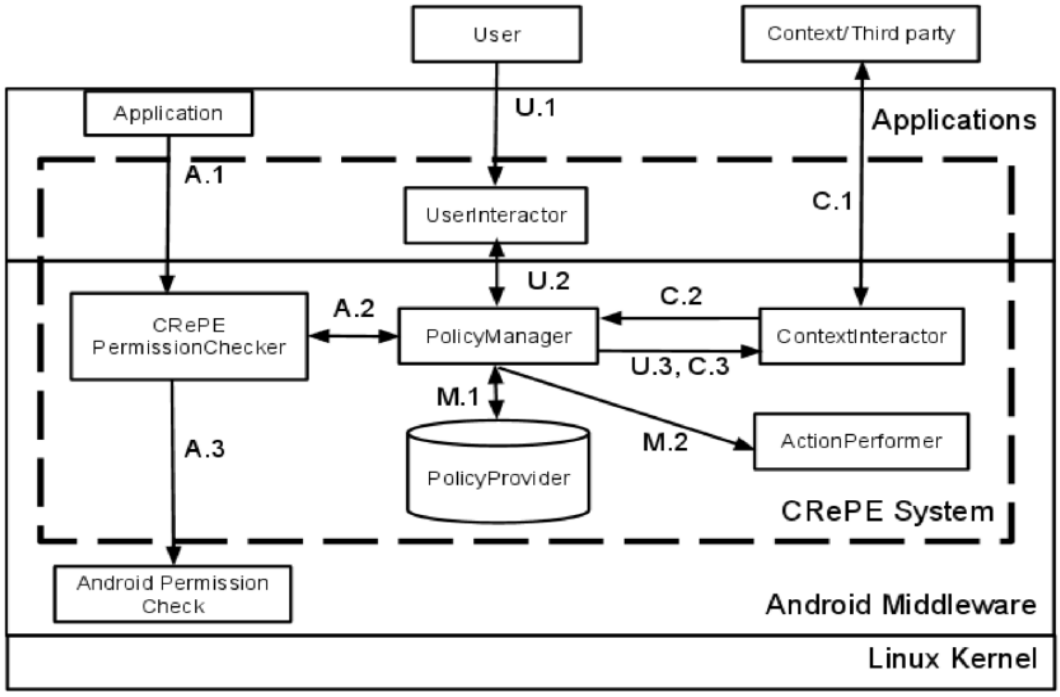


Fonte: LIU; NAM; SHIN, 2011, p. 1148.

4.6 **CRePE**

Em CRePE (CONTI; NGUYEN; CRISPO, 2011) é apresentada uma política de segurança baseada em contexto para o Android, que pode ser definido pelo status de algumas variáveis (ex: localização, tempo, temperatura, bateria etc.) ou pela combinação de variáveis. CRePE permite, assim, a definição de políticas de segurança refinadas definidas pelo usuário ou por terceiros, dependendo das autorizações.

Figura 13 – Arquitetura CRePE



O retângulo pontilhado representa todas as partes inseridas pelo mecanismo, incluindo a verificação de permissões, o gerenciamento de privacidade, o armazenamento de regras e o monitor de contexto.
O retângulo externo ao pontilhado representa o mecanismo existente na plataforma e as setas representam os desvios e a interação do CRePE com os mecanismos.

Fonte: CONTI; NGUYEN; CRISPO, 2011, p. 337.

A ideia é colocar a garantia da segurança antes da verificação de permissão do Android. Uma requisição de acesso é interceptada por CRePE, sendo que os componentes da solução são responsáveis por verificar se o acesso será permitido, baseado nas regras de contexto definidas.

A solução também envolve um monitor que informa quando um certo contexto é ativado ou desativado, utilizado para verificar quando há alterações no dispositivo. O usuário pode interagir com a solução através de uma interface gráfica em que os contextos podem ser definidos e armazenados.
A Figura 13 apresenta a alteração realizada na plataforma Android para a inserção do CRePE.

4.7 **Considerações finais**

O mecanismo aqui proposto pode ser considerado uma junção dos mecanismos apresentados anteriormente, assemelhando-se pelo fato de ter seu foco na privacidade dos usuários de dispositivos móveis e pela utilização de alguns conceitos em comum, como a mediação da verificação de acesso e a extensão do mecanismo de segurança presente em dispositivos móveis.

Capítulo 5

CAPÍTULO 5 - **MECANISMO DE PERSONALIZAÇÃO DA PRIVACIDADE**

Este capítulo apresenta a proposta do mecanismo para personalização da privacidade dos usuários de dispositivos móveis, tendo como intenção transferir o controle de acesso para o usuário através de regras personalizáveis, possibilitando configurar o acesso a recursos (hardware e conexões de rede) e as informações pessoais. O mecanismo aqui apresentado é genérico e pode ser implementado para qualquer sistema operacional.

5.1 **Mecanismo**

O Mecanismo para Personalização da Privacidade em Dispositivos Móveis (MPP), apresentado neste capítulo, tem como objetivo realizar a transferência do controle de acesso das aplicações instaladas em dispositivos móveis para o usuário. O mecanismo possibilita criar regras de acessos personalizadas para garantia da segurança e privacidade.

Atua realizando a mediação em tempo de execução de todos os acessos a hardware, dados e aplicações do dispositivo; a cada nova solicitação ocorre a intermediação, sendo esta concedida ou não de acordo com as regras personalizadas pelo usuário através de uma interface gráfica.

A proposta aqui apresentada estende o modelo existente de permissões do sistema operacional, possibilitando a criação de regras personalizadas pelo usuário de acesso a recursos. A aprovação da solicitação pelo mecanismo aqui proposto

implica sua aceitação pelo modelo já existente, pois a permissão foi concedida no momento da instalação da aplicação.

A principal diferença entre as propostas existentes, descritas no Capítulo 4, e o modelo apresentado neste trabalho está no fato de ele não apenas alterar o modelo existente nos sistemas operacionais executados em dispositivos móveis, mas também realizar a transferência do controle de permissões para o usuário, disponibilizando o controle sobre os acessos concedidos aos aplicativos de uma forma personalizável e trazendo maior segurança na forma de acesso ao hardware e aos dados dos usuários.

Com o intuito de fornecer uma maneira amigável para a criação das regras, a proposta utiliza uma interface gráfica através da qual é possível visualizar as permissões utilizadas pelas aplicações e definir regras de acesso personalizadas.

O mecanismo desenvolvido irá realizar a verificação das regras definidas pelo usuário antes de a solicitação ser enviada ao mecanismo existente na plataforma. Caso uma solicitação seja bloqueada pelo mecanismo, é apresentada a mensagem de bloqueio e a solicitação é interrompida, não sendo enviada ao mecanismo da plataforma. O principal objetivo das mensagens apresentadas pelo sistema é informar o usuário sobre o bloqueio.

A proposta aqui apresentada não altera a validação de segurança realizada quando se instala uma aplicação, o usuário continua tendo que conceder as permissões solicitadas quando está instalando uma nova aplicação.

Se julgar necessário, o usuário poderá utilizar a interface gráfica para alterar as permissões de acessos após a aplicação ter sido instalada.

A verificação de permissão é realizada através da mediação de todas as requisições ao *hardware* e aos dados.

Quando uma aplicação envia uma nova solicitação de acesso, o MPP realiza o desvio da solicitação e procura no repositório regras personalizadas pelo usuário para a permissão que está sendo solicitada e a aplicação que solicita.

Todas as regras definidas para a aplicação e o tipo de acesso serão verificadas. Caso alguma regra bloqueie o acesso, a requisição será interrompida.

Se o acesso requisitado pela aplicação for concedido pelo mecanismo, ou seja, não havendo qualquer regra personalizada pelo usuário bloqueando o acesso, a requisição será encaminhada para o modelo de verificação de permissões do

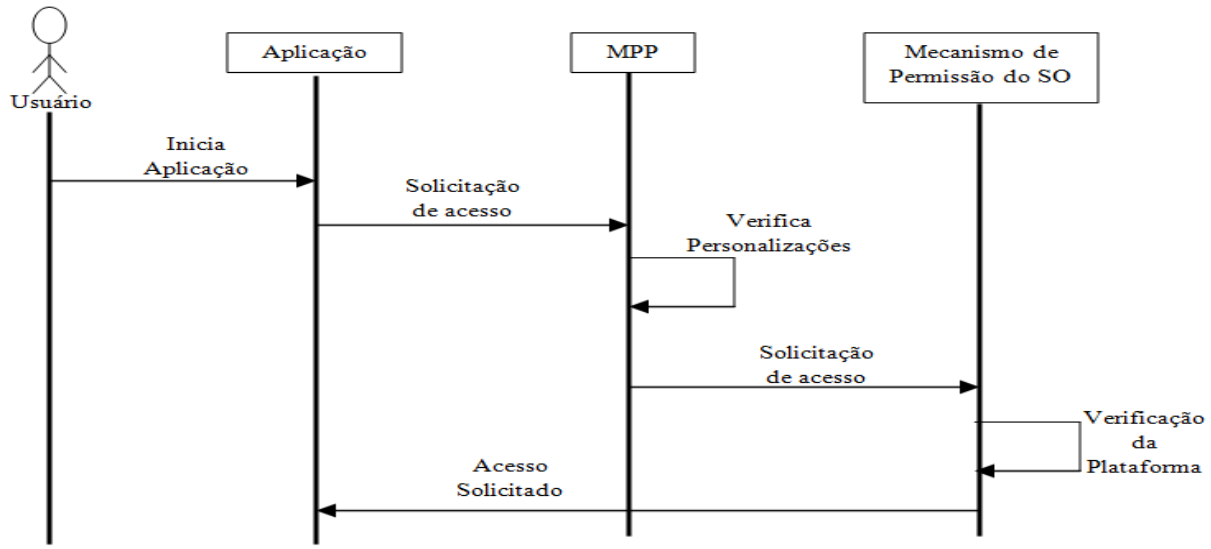
sistema operacional que aplicará as regras de acesso definidas na instalação da aplicação e disponibilizará o acesso para a aplicação que solicitou.

O protótipo desenvolvido realiza a validação de todas as solicitações incluindo as do sistema operacional. Na implementação existente, o mecanismo não obriga que o usuário defina uma regra para todas as permissões solicitadas pelas aplicações instaladas. Ou seja, o usuário pode escolher quais regras deseja personalizar.

A Figura 14 apresenta os diagramas com a sequência executada durante a validação das solicitações de acesso de uma requisição em relação à qual não foram encontradas regras que o bloqueiam .

Para este diagrama de sequência, o usuário solicitará o início da aplicação no momento em que a aplicação enviar a solicitação de acesso. Essa solicitação é mediada pelo mecanismo, que, nesse ponto, realiza o desvio da requisição para executar a consulta por regras personalizadas para a permissão que a aplicação solicita. Como nenhuma regra de bloqueio é encontrada, a solicitação é enviada para o mecanismo de segurança do sistema operacional que será responsável por retornar o acesso solicitado.

Figura 14 – Fluxo sem bloqueio

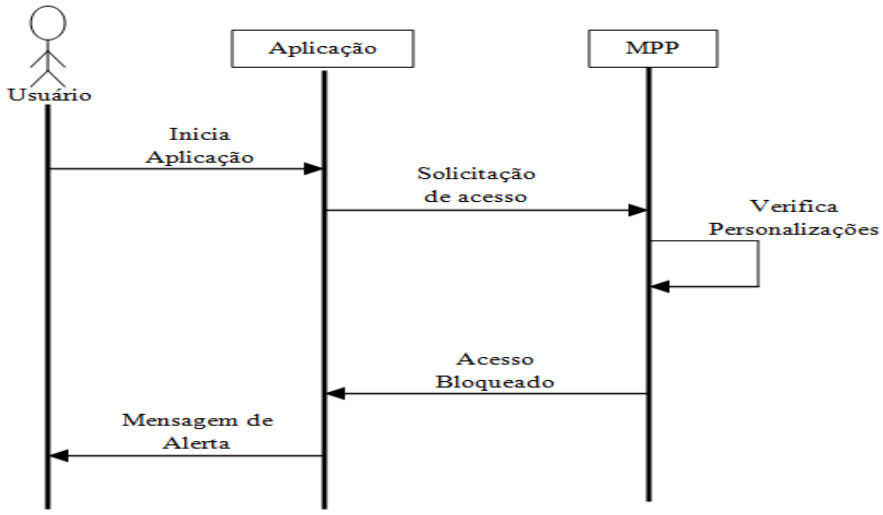


Fonte: Elaborado pelo autor.

A Figura 15 apresenta o sequência de execução quando um bloqueio definido pelo usuário é encontrado. A sequência a ser seguida é o mesma que a descrita n a

Figura 14, porém com uma diferença: como uma regra de bloqueio foi encontrada, a requisição não é enviada para o mecanismo do sistema operacional e uma mensagem é apresentada ao usuário.

Figura 15 – Fluxo com bloqueio



Fonte: Elaborado pelo autor.

Em decorrência dos bloqueios definidos nas personalizações feitas pelo usuário, podem ocorrer situações onde permissões concedidas no momento da instalação sejam revogadas, resultando no mau funcionamento de algumas aplicações.

Porém, como em aplicações para computadores pessoais, as desenvolvidas para dispositivos móveis também devem ser capazes de se recuperar de tais problemas. Falhas na execução de alguma aplicação devido às regras aplicadas pelo MPP serão apresentadas na forma de mensagens, assim como os bloqueios, com o intuito de informar o usuário, que deverá caso julgue necessário desbloquear o acesso ou tomar outra medida.

5.2 **Arquitetura**

A proposta foi desenvolvida utilizando o conceito de mediação total de requisições de acesso, conforme descrito no Capítulo 3: sempre que uma aplicação

solicita um acesso a um recurso, o mecanismo envia uma solicitação ao sistema operacional.

Nesse momento, o mecanismo realiza o desvio da solicitação para o módulo de verificação, o qual é responsável por extrair as informações necessárias das requisições para analisar a permissão de acesso.

Após extrair as informações necessárias, o mecanismo procura regras definidas e concede ou não o acesso baseado nas regras personalizadas pelo usuário e armazenadas pelo mecanismo.

Para a inserção das validações efetuadas pelo mecanismo, o modelo existente de permissões deve ser estendido para a inserção do desvio para o módulo responsável pela verificação das regras personalizadas, realizando assim a mediação total das solicitações de acesso em tempo de execução das aplicações.

Para realizar a verificação de permissão, a proposta desenvolvida utiliza-se de regras. Uma regra R=(<aplicação,componente>, <acesso>, <condição>) é uma definição de acesso personalizada pelo usuário através da interface gráfica, onde <aplicação,componente> define a aplicação e o componente para o qual uma regra se aplica, <acesso> *Sempre*|*Bloquear*|*Perguntar* define as permissões que a aplicação tem no componente e <condição> define a condição na qual a regra vai ser aplicada.

Um **acesso** define a permissão que uma aplicação tem sobre um componente como apresentado **Erro! Fonte de referência não encontrada.**.

Quadro 2 – Permissões de acesso

Permissão	Descrição
Sempre	O acesso deve ser liberado sempre que a condição definida pelo usuário for satisfeita.
Bloquear	Bloquear o acesso ao recurso sempre que satisfizer a condição definida pelo usuário.
Perguntar	Apresenta uma tela perguntando se o usuário permite o acesso.

Fonte: Elaborado pelo autor.

As condições foram inseridas no mecanismo para dar ao usuário a flexibilidade de personalizar a permissão de acesso de acordo com o contexto corrente do dispositivo.

A cada nova solicitação que for mediada, o mecanismo irá consultar as regras definidas pelo usuário para a aplicação As condições definidas para as regras serão avaliadas de acordo com o estado corrente do dispositivo e o acesso será liberado se todas as condições forem satisfeitas.

Uma ou mais variáveis podem ser adicionadas às condições. Isso é ilustrado nos exemplos de variáveis que podem ser utilizadas em uma condição no **Erro! Fonte de referência não encontrada.**.

Como, por exemplo, se a regra R for:

R=(<app1,INTERNET>,<Sempre>,<Wi-Fi>)

Quadro 3 – Exemplo de variáveis nas condições

Variável	Descrição
Bateria	Quantidade de bateria no dispositivo.
Rede	Tipo de rede a que o dispositivo está conectado.
Período	Intervalo de tempo no qual a permissão deve ser aplicada.
Tela	Se a tela está ativa ou não.
Localização	A localização geográfica do dispositivo.
Roaming	Se o dispositivo está em roaming.

Fonte: Elaborado pelo autor.

Tem-se que a app1 está permitida a acessar a *Internet* caso o dispositivo esteja conectado a uma rede Wi-Fi. Uma condição pode ser nula; neste caso a regra personalizada se aplicará sempre. O mecanismo deve informar ao usuário quando estiver definindo regras conflitantes.

Figura 16 – XML com regra personalizada

```
<regras component="Internet">
        <Acesso tupe="Sempre">
                <condicao>
                        "network=Wi-Fi"
                </condicao>
        </Acesso>
</regras>
```

Fonte: Elaborado pelo autor.

A Figura 16 apresenta um exemplo de uma regra personalizada. Regras definem o componente sobre o qual a regra será verificada, o tipo de acesso que será retornado e a condição sobre a qual será concedido tal acesso.

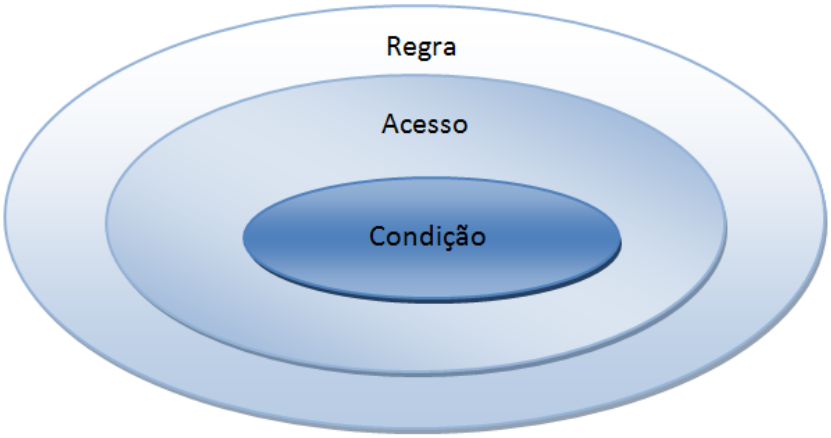
A Figura 17 apresenta a ilustração da composição das regras utilizadas pelo mecanismo, como por exemplo a apresentada na Figura 16. Esta demonstra a estrutura de uma regra personalizada utilizada pelo mecanismo.

As camadas de uma regra foram dispostas de forma que uma complete a outra formando assim uma regra válida para o mecanismo.

As camadas mais externas são compostas pelas camadas mais internas, responsáveis pela definição de detalhes da regra, como o tipo de acesso que a regra irá aplicar dada uma determinada condição.

Componentes são recursos para a criação de novos componentes ou outras aplicações, como a funcionalidade de procura na web. Com essa definição, neste trabalho, consideramos como um componente qualquer parte de outra aplicação ou do sistema operacional (hardware, conexão de rede e dados no dispositivo).

Figura 17 – Ilustração de uma regra



Fonte: Elaborado pelo autor.

Os conflitos ocorrem quando duas ou mais regras são definidas para a mesma aplicação e para o mesmo tipo de acesso, podendo levar a condições de indecisão em que o mecanismo não “saberia” se deve liberar ou não o acesso solicitado.

Um exemplo de conflito seria quando uma aplicação define uma regra "liberar sempre" e "bloquear" o acesso à *Internet* e ambas com condição nula. Duas ações

são possíveis simultaneamente. Uma forma de tornar o exemplo acima válido é a definição de uma condição como, por exemplo, bloquear se não estiver em uma rede wi-fi.

Com o intuído de sanar esse problema no protótipo implementado, foi realizada a restrição de possibilitar a criação de apenas uma regra para cada permissão solicitada pela aplicação.

A mediação é a captura da requisição no momento em que uma nova solicitação de acesso é enviada ao sistema operacional Essa solicitação é desviada para um módulo em que a verificação das regras personalizadas pelo usuário é realizada antes da liberação de qualquer acesso.

O foco do mecanismo não é simplesmente fornecer uma forma para que usuários de dispositivos móveis possam criar regras de acesso de aplicações instaladas. Pretende, também, fornecer um maior grau de segurança em certos aspectos não tratados pelo sistema operacional, como é o caso do compartilhamento de permissões por aplicações com a mesma assinatura, que é a situação da plataforma Android apresentada por Delac; Silic e Krolo (2011).

A proposta aqui apresentada não buscou a solução de todos os riscos à privacidade e segurança existentes em sistemas operacionais executando em dispositivos móveis, mas sim a disponibilização da funcionalidade não existente, como a de gerenciamento de permissões concedidas às aplicações.

5.3 **Implementação**

A ferramenta aqui apresentada buscou inovar a disponibilização da personalização da privacidade nos sistemas operacionais executados em dispositivos móveis.

Como estratégia de desenvolvimento do mecanismo, seguiu-se a proposta de arquitetura apresentada, realizando a extensão do modelo de verificação de permissões existentes no sistema operacional, permitindo a mediação total de acesso através da inserção de desvio da solicitação de acesso para o mecanismo, no qual esta será avaliada de acordo com as regras definidas pelo usuário.

Não foram realizadas alterações nas verificações de segurança no momento de instalação de uma nova aplicação. O usuário deverá continuar concedendo as permissões solicitadas pela aplicação no momento da instalação e, caso considere necessário, uma regra de acesso pode ser criada para controlar o acesso da aplicação.

O mecanismo foi desenvolvido de forma que as configurações definidas pelo usuário sejam verificadas antes de as solicitações serem enviadas para o mecanismo de verificação de permissões do sistema operacional.

Um ponto a se observar, e que foi avaliado, é o fato de que o modelo proposto não irá substituir o mecanismo de permissões existente, podendo assim gerar atraso devido às verificações.

Isso se deve ao fato de que a proposta aqui apresentada estende a validação de segurança existente e, mesmo que a requisição seja concedida, esta também será avaliada pelo mecanismo do sistema operacional executando no dispositivo.

Por meio da extensão do mecanismo de segurança existente, foi possível inserir a mediação capturando todas as solicitações de acesso realizadas para a utilização de recursos ou aplicações. Assim, as solicitações são desviadas para o mecanismo desenvolvido e avaliadas.

O mecanismo desenvolvido seguiu as definições de que mecanismos de segurança são ferramentas implementadas com o intuito de executar o controle físico e modelo de segurança computacional é um esquema para a especificação e aplicação de políticas de segurança.

Em geral, mecanismos realizam o controle baseados em um modelo de segurança pré-definido. Um mecanismo pode ser fundado sobre um modelo formal de direitos de acesso, de computação, de computação distribuída, ou mesmo em nenhum embasamento teórico específico.

Os princípios definidos por Saltzer e Schroeder e Denis e Van Horn para segurança apresentados no Capítulo 3 foram amplamente considerados durante o desenvolvimento do mecanismo, por serem referência e utilizados em sistemas computacionais, bem como na maioria dos mecanismos de segurança existentes nos sistemas operacionais executados em dispositivos móveis.

O Princípio de Economia de Mecanismo foi respeitado com a inserção do menor número de operações de código de computador para validar as solicitações e apenas um desvio foi inserido. Esse princípio é importante pois, quanto menor for o

mecanismo, menor será o impacto do mesmo sobre o sistema operacional. Quanto menor for a interferência do mecanismo, menor será o atraso inserido.

O mecanismo foi projetado para não conceder nenhum acesso sem o consentimento do usuário, estando de acordo com os Princípios de Falta de Acesso por Padrão e de Menor Privilégio. Todas as solicitações de acesso são verificadas, atendendo assim ao Princípio de Mediação Completa.

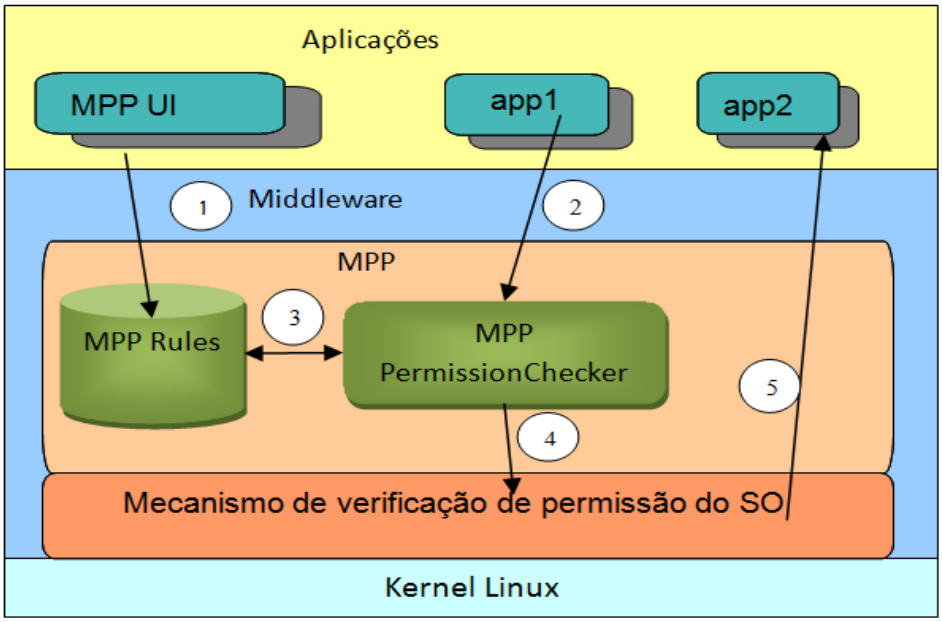
Os Princípios de Separação de Privilégios e de Projeto Aberto foram atendidos com a utilização de um controle de assinatura e a implementação de um mecanismo de código aberto.

O Princípio de Interceptar Todas as Requisições foi utilizado no desenvolvimento do mecanismo e sempre é invocado quando uma solicitação é realizada, atendendo, assim, o Princípio de Mediação Completa.

Com relação à aceitação psicológica, foi realizada uma avaliação de uso por meio de um estudo de campo. A avaliação consistiu na resposta a um questionário inicial, apresentação do mecanismo, execução de três casos de uso utilizando o protótipo implementado como prova de conceito e repostas a um segundo questionário.

O protótipo implementado seguiu a arquitetura previamente definida e é composto pelos três módulos MPP UI, MPP Rules e MPP PermissionChecker, como ilustrado na Figura 18 e descrito a seguir.

Figura 18 – Arquitetura MPP



Fonte: Elaborado pelo autor.

MPP UI é a interface desenvolvida como prova do conceito de criação de regras personalizáveis, por meio da qual o usuário é capaz de criar uma regra de acesso para cada permissão utilizada pela aplicação de acordo com suas necessidades.

Esse módulo foi desenvolvido como uma aplicação padrão e tem como objetivo disponibilizar um meio para que o usuário crie regras de acordo com o definido neste trabalho. Esse módulo também é responsável por tratar possíveis conflitos.

MPP Rules é o módulo do mecanismo responsável por armazenar as regras criadas por meio da MPP UI. Sua função é manter um repositório onde as regras são consultadas durante a verificação de acesso.

No protótipo desenvolvido, esse módulo foi implementado com um provedor de dados disponibilizado pela plataforma Android, por motivos de desempenho e facilidade de implementação.

Porém, para outras plataformas, podem ser utilizados arquivos XML para armazenar as regras. Algumas plataformas implementam APIs customizadas para trabalhar com esse formato de arquivos.

O mecanismo aqui apresentado utiliza um repositório de regras locais, porém pode ser alterado para a utilização de um repositório remoto de regras.

Por motivos de segurança, somente os componentes desenvolvidos para o mecanismo poderão acessar o repositório de regras. Visando garantir a segurança, foi utilizado o mecanismo de assinatura nativo da plataforma Android. No caso de regras definidas em repositórios externos, mecanismos de criptografia podem ser utilizados para garantir a segurança.

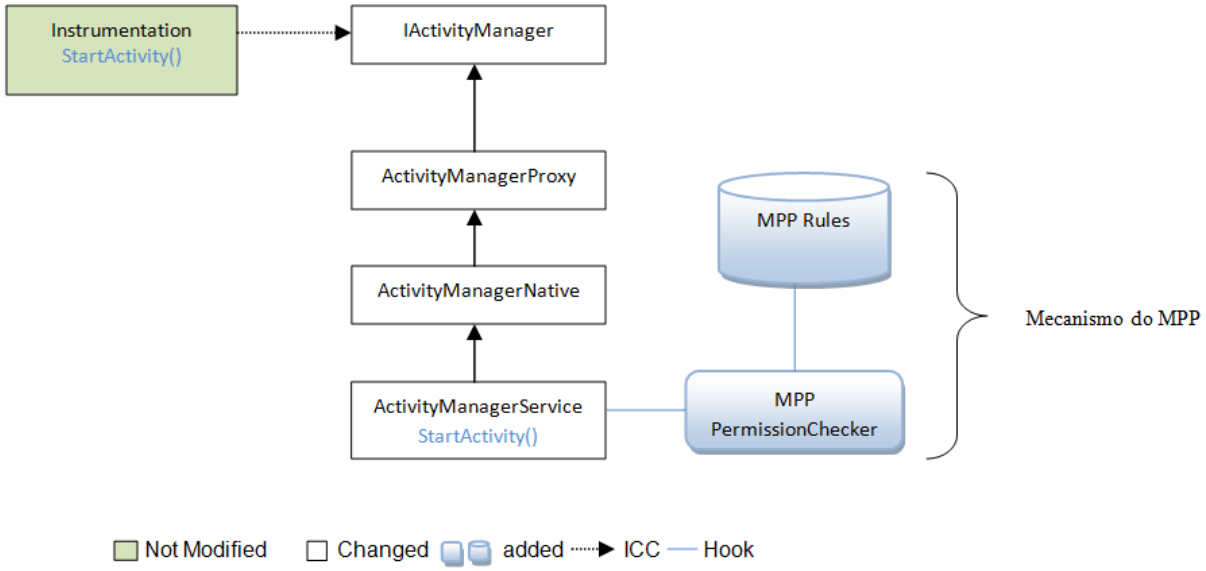
MPP PermissionChecker é a parte do mecanismo em que as verificações de fato acontecem. Após mediação de uma requisição de acesso, a solicitação é enviada para esse módulo, onde as informações necessárias para verificar o acesso são extraídas e em seguida é realizada a consulta no repositório de regras personalizadas do mecanismo MPP Rules para verificar a concessão ou não da permissão solicitada.

Para o desenvolvimento desse protótipo, duas partes existentes na plataforma Android foram alteradas com o intuito de disponibilizar todas as funcionalidades necessárias.

A primeira parte da plataforma alterada para a inserção do mecanismo, de maior importância, foram os módulos internos, responsáveis pela verificação das permissões no modelo existente, como apresentado na Figura 19.

As alterações foram relativas à inserção do desvio da requisição para o módulo MPP PermissionChecker. Como apresentado, esse é o responsável pela verificação de acesso através da análise dos dados da solicitação e do estado atual do dispositivo. Com essas informações, o mecanismo procura por regras personalizadas armazenadas no MPP Rules que bloqueiem o acesso solicitado.

Figura 19 – Alteração no mecanismo de segurança



Fonte: Elaborado pelo autor.

Para a implementação do mecanismo na plataforma Android, foi realizada a alteração da classe ActivityManagerService no método checkComponentPermission, responsável por realizar as verificações de permissões dos componentes da plataforma, sejam estes disponibilizados com a plataforma ou criados por terceiros.

A

Figura 20 apresenta a implementação da verificação da permissão, mais especificamente o desvio criado para chamar o mecanismo aqui proposto. Nesse ponto, o mecanismo realiza a mediação da solicitação de acesso com a chamada do MPP PermissionChecker, onde serão extraídas todas as informações necessárias para a verificação das permissões.

Figura 20 – Desvio para chamada ao mecanismo

```
int checkComponentPermission(String permission, int pid, int uid, int owningUid, boolean exported) {
    // We might be performing an operation on behalf of an indirect binder
    // invocation, e.g. via {@link #openContentUri}.  Check and adjust the
    // client identity accordingly before proceeding.
    //parte alterada para a inserção da proposta
    if(permission!=null ){
        try {
            // recupera o nome da aplicação que esta solicitando a permissão
            String [] st = AppGlobals.getPackageManager().getPackagesForUid(uid);
            if(st != null){
                MppControler mpp = new MppControler ();
                for (int i = 0 ; i < st.length; i++){
                    // verifica se existe um bloqueio definido para a aplicação e a regra sendo solicitada
                    if (mpp.checkPermission(mContext.getContentResolver(),permission, st[i] ) == PackageManager.PERMISSION_DENIED){
                        // monta a mensagem a ser enviada
                        Message msg = Message.obtain();
                        HashMap map = new HashMap();//utilizado para enviar o nome da aplicação e a permissão como parametro
                        msg.what = MPP_MSG; // mensagem a ser enviada
                        msg.obj = map; // parametros
                        map.put("app", st[i]);//nome da aplicação
                        map.put("permission", permission );//nome da permissão
                        //envia a mensagem de bloqueio
                        mHandler.sendMessage(msg);
                        // envia a mensagem de bloqueio
                        return PackageManager.PERMISSION_DENIED;
                    }
                }
            }
        } catch (RemoteException e) {
            e.printStackTrace();
        }
    }
    //Fim da parte alterada para a inserção da proposta
```

Fonte: Elaborado pelo autor.

Os atributos enviados como parâmetros são utilizados pelo mecanismo durante a validação do acesso.

Se for encontrada alguma regra de bloqueio, a requisição não será enviada para o mecanismo do Android e o acesso ao recurso será negado, bem como será emitida a mensagem de bloqueio.

A **Erro! Fonte de referência não encontrada.** apresenta a implementação do mecanismo de verificação de segurança. Quando o desvio para o método responsável por executar a verificação da permissão é realizado – mais especificamente o método checkPermission do módulo MPP PermissionChecker – este recebe como parâmetro um resolvedor de conteúdo, a permissão que está sendo solicitada e a aplicação solicitante.

O resolvedor de conteúdo é importante por ser o responsável por executar a consulta das regras na base de armazenamento de regras MPP Rules e os demais parâmetros são utilizados na condição para refinar a consulta.

Figura 21 – Verificação da permissão pelo mecanismo

```
public int checkPermission(ContentResolver db,String permission, String app){
    //String com clausula que será utilizana na consulta
    String selectionClause = "aplicacao = ? and componente = ?";
    //monta os parametros que seram utilizados na consulta
    String[] selectionArgs = new String[2];
    selectionArgs[0] = app;//nome da aplicação
    selectionArgs[1] = permission;//nome da permissão
    Cursor regra = null;//Cursor utilizado para armazenar o retorno da consulta

    //Consulta por regras customizadas
    try{
        regra = db.query(Uri.parse("content://br.com.mpp.contentprovider.RegrasContentProvider/regras"), null, selectionClause, selectionArgs, null);
    } catch (Exception e) {
        e.printStackTrace();// emite erro caso a consulta falhe
    }
    //Verifica se alguma regra personalizada foi encontrada a permissão de acesso
    if(regra != null){
        if ( regra.moveToFirst() ){
            //Verifica se existe bloqueio definido
            if(regra.getString( regra.getColumnIndex("tipo")).equals("Bloquear") || regra.getString( regra.getColumnIndex("tipo")).equals("Block")){
                regra.close();//fecha o recurso utilizado na consulta
                //envia a mensagem de bloqueio
                return PackageManager.PERMISSION_DENIED;
            }
        }
     regra.close();//fecha o recurso utilizado na consulta
     return 0;//não há bloqueios.retorna para a execução normal
    }
    return 0; //não há regras definidas, retorna para a execução normal
}
```

Fonte: Elaborado pelo autor.

Com essas informações, é realizada uma consulta por regras personalizadas pelo usuário. Caso alguma regra de bloqueio seja encontrada, o mecanismo envia um sinal de permissão negada à plataforma e a solicitação é interrompida. Se não forem encontradas regras que bloqueiem a permissão solicitada, a requisição será enviada à plataforma e seguirá o fluxo normal.

No modelo de permissões existente no Android, somente as aplicações são capazes de verificar as permissões concedidas no momento da instalação. Como o mecanismo foi implementado como parte do núcleo da plataforma e mantém seu

próprio repositório de regras, ele é capaz de intermediar e verificar a permissão baseado nas personalizações realizadas pelo usuário.

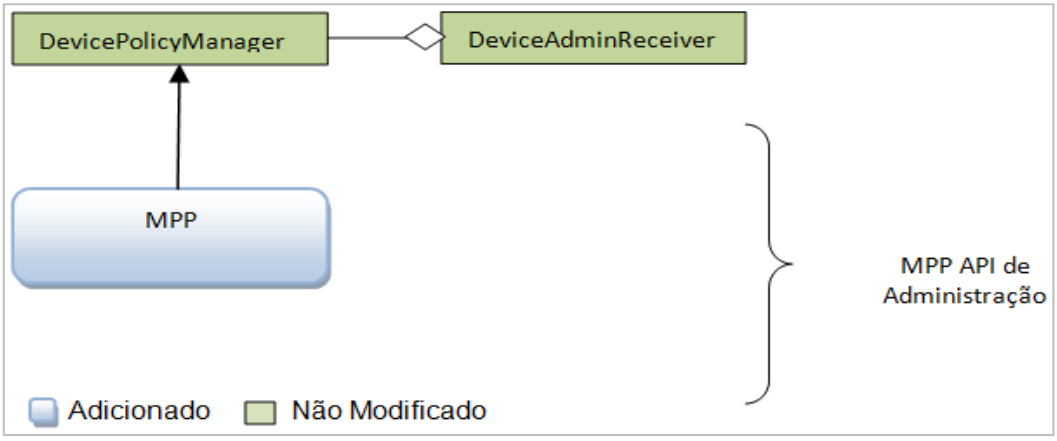
Outra parte da plataforma Android alterada foi a extensão da API (*Applicatom Program Interface*) de administração, responsável pelas funcionalidades administrativas do dispositivo disponibilizadas pela plataforma Android, a partir da versão 2.2.

Essa API, como já apresentada no Capítulo 3 sobre o Android, tem o intuito de oferecer uma interface para o gerenciamento da segurança dos dispositivos. Tendo o mecanismo aqui proposto o mesmo objetivo, a API foi estendida para oferecer algumas funcionalidades adicionais.

A API existente tem a função de fornecer algumas funcionalidades para que empresas possam validar as senhas que os usuários escolhem para seus dispositivos e a encriptação de alguns dados.

As extensões inseridas tiveram a finalidade de oferecer a interface gráfica funcionalidades para consultar, inserir, alterar e apagar regras armazenadas no repositório. A extensão realizada é apresentada na Figura 22.

Figura 22 – Extensão da API de Administração



Fonte: Elaborado pelo autor.

Não foram realizadas mudanças dos métodos existentes na API, mas sim a inserção de novas funcionalidades que permitem que a aplicação de configuração implementada como prova de conceito interaja com o repositório do mecanismo.

A Figura 23 mostra a classe que foi criada para estender a API de administração existente.

Figura 23 – Extensão da classe de administração do Android

```
public class MPPPoliceManager extends DevicePolicyManager {
```

Fonte: Elaborado pelo autor.

A Figura 24 apresenta o método Add criado para adicionar uma nova regra ao repositório do mecanismo através da interface. Como a implementação do tratamento de conflito foi feita possibilitando a configuração de apenas uma regra para cada permissão, no momento em que vai salvar a regra, esse método procura por regras existentes e, caso encontre, atualiza a regra de acordo com a nova personalização. Caso não seja encontrada uma regra já definida para a aplicação, uma nova é criada.

Figura 24 – Método Add criado para adicionar uma nova regra

```
public void add(View view)
{  // recupera os parametros para criar a regra inseridos pelo usuário na interface
    ContentValues values = new ContentValues();
    values.put( Regra.Regras.APLICACAO, this.pacote );
    Spinner permissao = (Spinner) findViewById(R.id.permissao);
    values.put( Regra.Regras.COMPONENTE, permissao.getSelectedItem().toString());
    Spinner tipoAcesso = (Spinner) findViewById(R.id.tipoAcesso);
    values.put( Regra.Regras.TIPO, tipoAcesso.getSelectedItem().toString() );
    EditText condicao = (EditText) findViewById(R.id.condicao);
    values.put( Regra.Regras.CONDICAO, condicao.getText().toString() );

    //clausula que será utilizada na consulta
    String selectionClause = selectionClause = Regra.Regras.APLICACAO + " = ? and " + Regra.Regras.COMPONENTE + " = ? ";

    // Cria o array que terá os parametros da consulta
    String[] selectionArgs = new String[2];
    selectionArgs[0] = this.pacote; // mo,e da aplicação
    selectionArgs[1] = permissao.getSelectedItem().toString(); // nome da permissão

    //procura por regras ja existentes para a aplicação e a permissão sendo consedida
    Cursor regrasPers = getContentResolver().query( Regra.Regras.CONTENT_URI, null, selectionClause, selectionArgs, null );
    if(regrasPers.getCount() == 0){ // não existe regra definida para a combinação
        //insere a condição na base de dados
        Uri uri = getContentResolver().insert( Regra.Regras.CONTENT_URI, values );
    }else{
        //atualiza a regra ja existente na base
        regrasPers.close();
        getContentResolver().update( Regra.Regras.CONTENT_URI, values,selectionClause,selectionArgs );
    }

    // cria um ArrayAdapter utilizando uma lista.
    MPPRegraArrayAdapter listAdapter = new MPPRegraArrayAdapter(this, R.layout.simple_row,  getPermission(pacote));

    // seta o ArrayAdapter como adaptador de um ListView's.
    ListView rolesListView = (ListView) findViewById( R.id.roles );
    rolesListView.setAdapter( listAdapter );
}
```

Fonte: Elaborado pelo autor.

Uma das maiores vantagens da extensão dessa API é que usuários e empresas poderão criar suas próprias aplicações. Como o mecanismo implementado foi inserido diretamente no núcleo da plataforma, a incorporação de novas funcionalidades na API de gerenciamento também possibilita a criação de interfaces de controle para o mecanismo por outras pessoas.

Por exemplo, uma aplicação definida por uma empresa poderá ler as configurações dos dispositivos de seus funcionários, criar um repositório de regras centralizado e aplicar essas regras de segurança a todos os dispositivos, possibilitando, assim, um gerenciamento da segurança dos mesmos de forma não intrusiva.

5.4 **Protótipo**

Com o intuito de provar a funcionalidade da proposta aqui apresentada e avaliar algumas de suas características, como o desempenho e a aceitação por usuários de dispositivos móveis, foi implementado um protótipo do mecanismo utilizando a plataforma Android.

Conforme exposto anteriormente, o motivo da escolha da plataforma para a implementação da prova de conceito se deu pelo fato de que a plataforma é de código aberto e fornece facilidades para alterações e geração de versões que contenham as alterações.

O código base utilizado na implementação e alterado para a inserção do mecanismo é da versão 4.0.3 da plataforma e foi obtido através do repositório disponibilizado pelo Google na *Internet*.

Como plataforma de desenvolvimento, foi utilizado o sistema operacional Ubuntu 12, escolhido pelo fato de a ferramenta disponibilizada pela plataforma Android para a compilação de novas versões só executar em plataformas Linux e MAC.

Para a alteração do código fonte e a depuração das alterações foi utilizada a plataforma Eclipse juntamente com o *plugin* SDK (*Sistem Develop Kit*), também disponibilizado pelo Google.

Essas ferramentas foram de grande utilidade durante o desenvolvimento do protótipo, pois disponibilizam funcionalidades para monitorar e depurar as alterações realizadas. O mecanismo foi implementado como descrito no capítulo 5.3; para tanto, dois módulos foram implementados.

O primeiro deles, no núcleo de verificação do Android, com a inserção do desvio para as classes que realizam a implementação do modelo e os métodos para acesso do mecanismo através da interface gráfica. No momento desse desvio, o mecanismo realiza a mediação da solicitação. Essa fase foi inserida antes da verificação de permissão da plataforma.

A implementação do mecanismo teve como foco o menor número de operações para realizar uma verificação, a fim de diminuir o atraso ao verificá-la.

Após o mecanismo realizar a mediação da solicitação, é realizada a chamada ao mecanismo que realiza a verificação das regras.

Nesse momento foram utilizadas funcionalidades disponibilizadas pela plataforma, como a execução de pesquisa de dados em um repositório, para realizar a consulta por regras personalizadas. Algumas regras de exclusão foram empregadas para melhorar o desempenho do mecanismo, como, por exemplo, a exclusão da verificação dos componentes utilizados durante a inicialização e usados pelo mecanismo em tempo de execução.

Com base no retorno da consulta ao repositório do mecanismo, é verificado se um bloqueio foi definido. Em caso afirmativo, um sinal de bloqueio é enviado como retorno e a solicitação é interrompida. Já em caso de concessão da permissão, a solicitação é enviada para o mecanismo e continua seu fluxo normal.

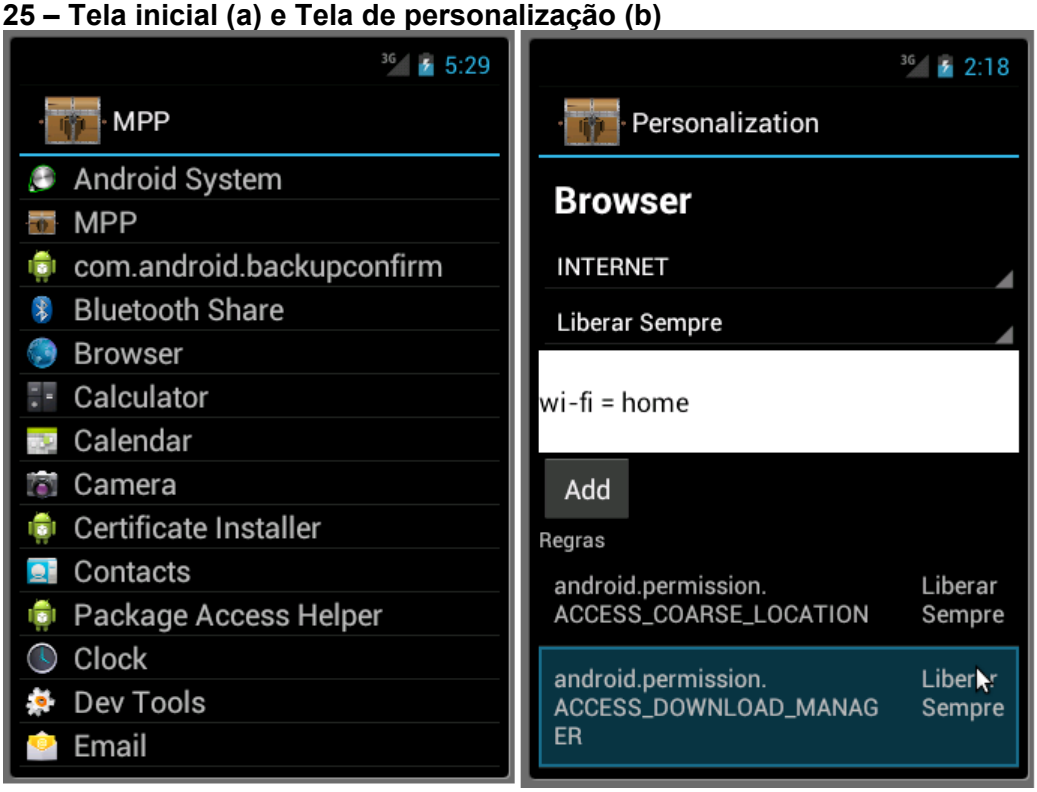
A interface implementada para o gerenciamento do mecanismo é apresentada na

Figura 25. Essa interface apresenta todas as aplicações instaladas no dispositivo -

Figura 25(a) - e ao selecionar uma aplicação é possível ver todas as permissões solicitadas e criar regras personalizadas -

Figura 25(b). O mecanismo, como descrito acima, realiza a verificação de permissões de forma dinâmica e baseada no estado do dispositivo. Após a criação de uma nova regra, esta já será aplicada na próxima vez que a aplicação solicitar a permissão.

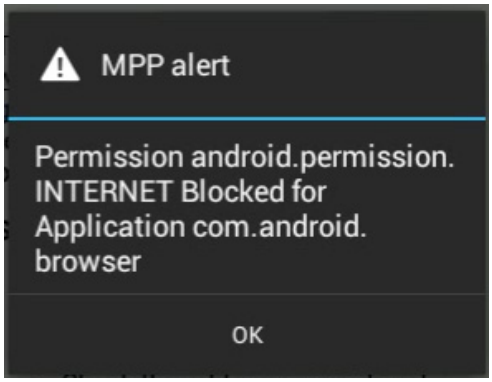
Figura 25 – Tela inicial (a) e Tela de personalização (b)



Fonte: Elaborado pelo autor.

A Figura 26 mostra um exemplo da mensagem apresentada pelo mecanismo ao usuário no momento em que um bloqueio é realizado. Desta forma, se o usuário julgar necessário, poderá alterar a regra definida.

Figura 26 – Alerta de bloqueio



Fonte: Elaborado pelo autor.

O protótipo foi utilizado para a realização de uma avaliação de uso com o intuito de mensurar a aceitação e facilidade de uso; testes para verificar os atrasos inseridos pela proposta também foram executados e os resultados serão apresentados no próximo capítulo.

5.5 **Considerações Finais**

O mecanismo aqui apresentado é uma combinação das técnicas abordadas nos trabalhos relacionados no Capítulo 4, no que diz respeito a interceptar e validar requisições, porém utilizando uma metodologia diferenciada.

O maior diferencial do mecanismo aqui proposto consiste no fato de realizar a transferência do controle de personalização do acesso para o usuário por meio de regras e não simplesmente inserir um novo método para avaliar as permissões.

A implementação do modelo mostrou-se viável, sendo capaz de fornecer uma maior segurança aos usuários de dispositivos móveis, como será apresentado no próximo capítulo.

Entre as principais contribuições da proposta aqui apresentada destacam-se a disponibilização de uma maneira de controlar o modo como as aplicações utilizam o dispositivo, bem ou aumento da consciência do usuário sobre o que cada aplicação realiza.

A plataforma Android utilizada mostrou-se satisfatória, possibilitando a alteração para a inserção do mecanismo e auxiliando muito no seu processo de desenvolvimento, destacando-se as funcionalidades para criar novas versões customizados.

Entretanto, como já dito, o mecanismo foi desenvolvido para ser aplicável a todos os sistemas operacionais executados em dispositivos móveis, tendo sido a prova de conceito da proposta criada para Andorid devido ao fato de o código fonte dos demais sistemas operacionais para dispositivos móveis não estarem disponíveis para acesso público.

Durante a elaboração deste trabalho, foram pesquisadas maneiras de implementar a proposta para outras plataformas, esbarrando porém na dificuldade de acesso ao código dos demais sistemas operacionais e em restrições, como a necessidade do dispositivo do fabricante para criar ferramenta, como é o caso dos produtos da Apple.

Limitaram a evolução do estudo nesse sentido. uma solução que poderia sanar essa dificuldade seria o contato com os fabricantes para verificar possíveis alternativas de testar a proposta.

Devido à limitação, o que pode ser feito para contribuir com a evolução do estudo com relação a implementação foi a pesquisa da especificação das demais plataformas, como iOS e proposição de locais onde o sistema operacional poderia ser alterado para a inserção da proposta. Uma funcionalidade adicional que poderia ser oferecida pelo mecanismo e que garantiria a segurança do usuário.

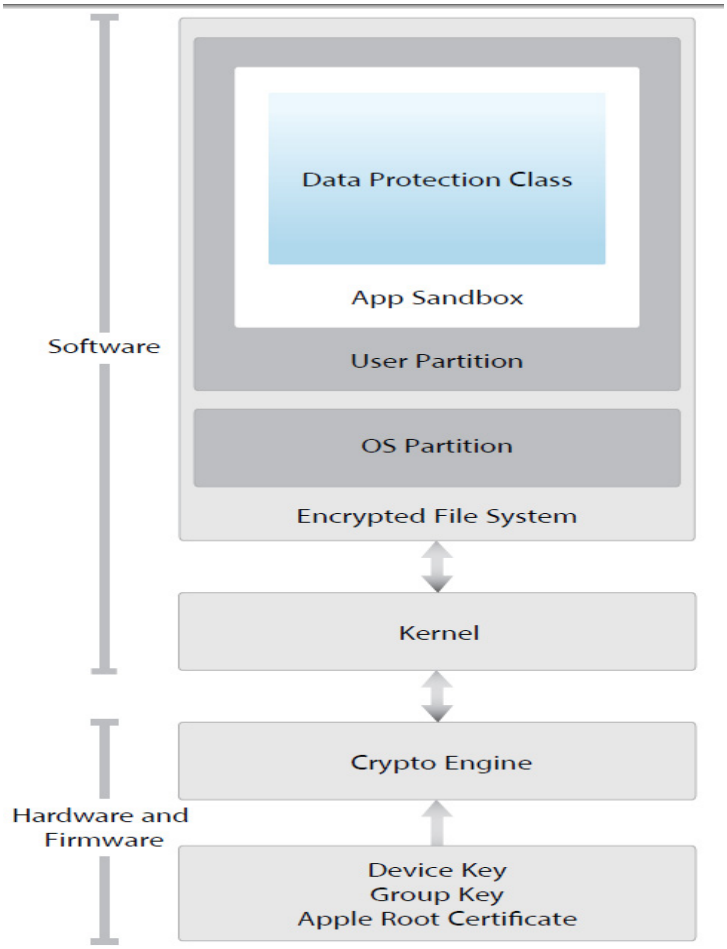
Seria a definição de um alerta, ou uma tela pop-up, apresentada ao usuário perguntando se deseja conceder permissão quando alguma solicitação de acesso sem regra personalizada for detectada. Assim, o recurso só seria disponibilizado se o usuário permitisse manualmente.

Trabalhos futuros, visando à implementação do mecanismo sem a necessidade de alteração do núcleo do sistema operacional executado no dispositivo, podem auxiliar sua aceitação e instalação em outros SO para dispositivos móveis.

A

Figura 27 apresenta o modelo de arquitetura de segurança existente na plataforma iOS (APPLE, 2013). Um possível ponto de alteração da plataforma para a inserção do mecanismo proposto seria na camada de *User Portion,* por ser a camada responsável pela segurança do usuário e o mecanismo aqui desenvolvido se propor a transferir o controle de permissão para o usuário. Porém, não foi possível avançar na avaliação dos pontos onde as alterações deveriam ser realizadas, por não estar o código disponível para análise.

Figura 27 – Modelo de Segurança iOS



Fonte: Apple, 2013.

Capítulo 6

CAPÍTULO 6 - **EXPERIMENTOS E RESULTADOS**

Este capítulo apresenta os experimentos realizados com o intuito de verificar algumas características do mecanismo proposto, como aceitabilidade, facilidade de uso e o atraso provocado, bem como os resultados obtidos.

6.1 **Avaliação de Uso**

De acordo com Holzinger (2005), deve-se conduzir uma observação de campo quando se foca na usabilidade, que tem como objetivo identificar problemas óbvios que podem impactar o produto.

O estudo de campo consiste na observação de um ou mais participantes durante a utilização do software que se pretende avaliar, registrando os fatos observados em notas, vídeos ou registros textuais. Métodos auxiliares como questionários também podem ser utilizados para ajudar na avaliação.

De acordo com Yin (2005), estudos de Caso são empregados quando se quer investigar e compreender em profundidade fenômenos sociais através de abordagens empíricas e holísticas de problemas contemporâneos.

Seu objetivo é compreender o evento em estudo e ao mesmo tempo desenvolver teorias mais genéricas a respeito dos aspectos característicos do fenômeno observado (FIDEL, 1993). A abordagem de estudo de caso não é um método propriamente dito, mas uma estratégia de pesquisa.

Para Trauth e O'Connor (1991), um dos principais benefícios do estudo de caso é sua característica de ser dirigido aos estágios de exploração, classificação e desenvolvimento de hipóteses. Normalmente, mais de uma entidade (pessoa, grupo, organização) são examinadas.

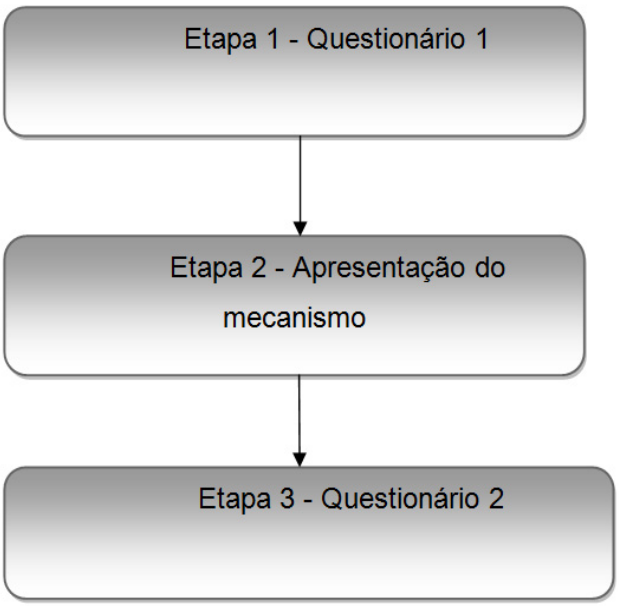
Com o intuito de mensurar a aceitação e a facilidade de uso do protótipo implementado para o mecanismo de personalização de privacidade proposto neste trabalho (MPP), foi realizada uma observação de campo.

O estudo foi composto por três cenários em que os participantes utilizavam um dispositivo com e sem a proposta instalada e forneciam resposta às perguntas de dois questionários.

6.1.1 Metodologia

A metodologia utilizada para a avaliação de uso do mecanismo aqui proposto foi a observação de campo, dividida em três fases, como ilustrado na Figura 28.

Figura 28 – Etapas do avaliação de uso



Fonte: Elaborado pelo autor.

Com o intuito de obter informações complementares, dois questionários foram elaborados, com itens para medir o grau de segurança dos participantes ao utilizarem um dispositivo com e sem uma solução para o controle da privacidade e segurança instalada.

Também com esse objetivo, cinco afirmativas idênticas foram inseridas em ambos os questionários. O objetivo dessas afirmativas foi a realização de uma

avaliação estatística no sentimento de segurança com e sem a proposta instalada no dispositivo.

As afirmativas são apresentadas abaixo:

1. Considero importante poder configurar as permissões das aplicações instaladas no dispositivo.
2. Confio na forma com que as aplicações utilizam os recursos do dispositivo.
3. Sinto-me seguro quando utilizo meu dispositivo móvel.
4. Considero que os dados em meu dispositivo estão seguros.
5. Sinto-me seguro com relação aos dados armazenados em meu dispositivo móvel.

Na primeira fase do estudo, foi entregue aos participantes um questionário denominado Questionário 1, com o objetivo de identificar o grau de segurança que os mesmos acreditam ter ao usar seus dispositivos, bem como apurar o grau de conhecimento sobre a plataforma que utilizam.

A coleta destas informações, antes da apresentação do mecanismo, é importante para a comparação estatística das respostas após a apresentação e utilização do mecanismo pelos participantes.

Na segunda etapa, o mecanismo proposto foi apresentado aos participantes, bem como os riscos que este visa afastar. Essa apresentação foi realizada com o emprego de slides explicativos (Anexo V) e de um vídeo de demonstração da utilização do protótipo. Tais apresentações foram realizadas pelo pesquisador Leonardo Leite de Melo, responsável pelo estudo.

Na terceira etapa, três cenários de uso do dispositivo foram entregues aos participantes. Cada um dos três cenários continha informações para os participantes utilizarem o dispositivo normalmente, sem bloqueios definidos, e, em seguida, utilizar o mecanismo aqui proposto para bloquear alguma permissão de acesso. Depois, utilizaram novamente as aplicações para verificar a consequência do bloqueio personalizado.

Essa etapa teve como objetivo o contato dos participantes com o mecanismo e a utilização do mesmo, para que fossem capazes de responder ao segundo questionário.

Esse segundo questionário, denominado, Questionário 2, teve como objetivo obter informações sobre a aceitação, a facilidade de uso e a segurança proporcionadas pelo mecanismo. As perguntas presentes nos dois questionários

objetivavam a comparação das respostas antes e após o uso do mecanismo proposto.

O estudo de campo foi realizado com 20 participantes. As respostas fornecidas pelos participantes para cada pergunta tinham um valor da escala de Likert, com possibilidade de comentários pelo participante.

A escala de Likert é um tipo de escala psicrométrica em geral quando se tem intuito de medir as atitudes ou o comportamento. Esse tipo de escala é especialmente útil quando se deseja coletar informações sobre assuntos sensíveis ou desafiadores.

A escala se propõe a realizar tal medição utilizando opções de respostas que variam de um extremo a outro (ex: de concordo totalmente a discordo totalmente).

Ao fornecerem respostas baseados nessa escala, os perguntados informam sua concordância com uma afirmação. Ao contrário de questionários com resposta simples "sim e não", os que utilizam a escala de Likert permitem avaliar o nível da opinião.

Deve-se manter uma distinção entre a escala de Likert e um item de Likert. Um item é apenas um afirmação seguida de uma escala visual (ex: uma linha visual em que o participante indica sua resposta).

Para a elaboração de cada item do questionário, devem-se criar afirmações que têm como intuito medir a atitude com relação ao que se pretende avaliar. A escala é a soma das respostas fornecidas a cada item.

Existe muita polêmica entre os pesquisadores com relação ao nível fato de as respostas utilizarem uma escala de cinco ou sete pontos. Os críticos alegam que uma escala de cinco pontos, a mais habitual, pode não fornecer precisão nos intervalos que se pretende observar. Em geral o que é adotado é uma escala de maior nível sete, nove ou até mais pontos de acordo com a precisão que se busca.

Outro ponto de discordância entre os pesquisadores é com relação à resposta central ao questionário, que pode ser considerada como uma opinião indiferente por parte do respondente. Porém o uso de métodos estatísticos podem fornecer uma solução para esse problema.

Ao avaliar os dados desse tipo de escala, pode-se fazer dois tipos de análise: ordinal ou por intervalo, dependendo da forma que se considera a escala. Essa definição é essencial pois valores ordinais não podem ser analisados por métodos estatísticos como média ou desvio.

As principais vantagens das escala de Likert em relação às outras, segundo Mattar (2001) são a simplicidade de construção; o uso de afirmações que não estão explicitamente ligadas à atitude estudada, permitindo a inclusão de qualquer item que se verifique, empiricamente, ser coerente com o resultado final; e, ainda, a precisão de informação da opinião do respondente em relação a cada afirmação que a amplitude de respostas permitidas apresenta.

Como desvantagem, por ser uma escala essencialmente ordinal, não permite dizer quanto um respondente é mais favorável que outro, nem mede o quanto de mudança ocorre na atitude após expor os respondentes a determinados eventos.

Devido aos fatos apresentados, fez-se a escolha de uma escala de Likert de sete pontos, pois se busca uma maior granularidade nas respostas fornecidas com o intuito de medir a alteração na atitude dos participantes em dois momentos diferentes.

O Quadro 4 apresenta o significado de cada valor da escala que os participantes poderiam escolher, previamente apresentado.

Quadro 4 – Quadro de significado da respostas para a escala de sete pontos

Valor	Significado
1	Concordo Fortemente
2	Concordo
3	Concordo Parcialmente
4	Indiferente
5	Discordo Parcialmente
6	Discordo
7	Discordo Fortemente

Fonte: Elaborado pelo autor.

Para a validação dos resultados, os dados foram interpretados como intervalos e avaliados estatisticamente utilizando o teste T de Student, que se adequa aos dados analisados e à avaliação pretendida – uma mesma população em momentos diferentes

A avaliação de campo foi realizada no Laboratório do Departamento de Computação da Universidade Federal de São Carlos. Para a interação dos participantes, foi utilizado um ambiente de simulação da plataforma Android com o

núcleo da plataforma alterado para a inserção do mecanismo e da interface gráfica que fazem parte do protótipo.

O código fonte inicial do sistema Android foi obtido dos repositórios disponibilizados pelo Google versão 4.0.3 e foi alterado para a inserção do mecanismo, compilado e instalado nos simuladores que os participantes utilizaram.

6.1.2 Avaliação dos resultados

Uma observação de campo com o intuito de realizar a avaliação do uso de um software é especialmente importante quando se deseja analisar o comportamento dos usuários. Os resultados aqui apresentadas foram baseados na observação de campo e nas respostas aos Questionários 1 e 2 fornecidas pelos participantes no estudo.

Essa avaliação é necessária uma vez que o mecanismo requer algumas configurações para realizar bem seu funcionalidade – no caso deste estudo, a configuração de regras de acesso – e sua aceitação pelo usuário final é um dos pontos mais importantes para que o mecanismo seja efetivo, conforme Princípios de segurança apresentados no Capítulo 3.

.

Com o intuito de mensurar os pontos já apresentados, os questionários desenvolvidos para os estudo foram elaborados de forma a conter três grupos de afirmativas, o primeiro para se obter o conhecimento atual dos participantes sobre a plataforma, o segundo para medir a satisfação dos usuários com a proposta implementada e um terceiro para medir o sentimento de segurança ao utilizar o dispositivo com e sem a proposta instalada.

Com o objetivo de mensurar o sentimento de segurança com e sem a proposta instalada, cinco perguntas repetidas foram inseridas em ambos os questionários

Para uma melhor análise dos resultados da pesquisa, foi realizada uma abordagem quantitativa para estabelecer a média ponderada (MP) de cada resposta para o questionário utilizando uma escala de Likert de sete pontos apresentada no Quadro 4, com o objetivo de mensurar o grau de concordância dos participantes que responderam aos questionários.

A verificação quanto à concordância ou discordância das questões avaliadas, foi realizada obtendo-se a MP atribuída às respostas, relacionando à frequência das respostas dos entrevistados que fizeram tal atribuição.

A média ponderada é calculada através do somatório das multiplicações entre valores e pesos divididos pelo somatório dos pesos.

Considerou-se que valores inferiores a quatro são concordantes e maiores que quatro, discordantes, sendo a escala de sete pontos. O valor exato de quatro seria considerado "indiferente", "sem opinião" ou "neutro", equivalente aos casos em que os entrevistados deixaram em branco.

A equação MP (1) foi utilizada para avaliar todas as respostas. Nessa equação, "Si" representa a quantidade de vezes que o valor da escala de Likert foi selecionado como resposta e os números ordinais o valor atribuído a cada resposta.

Com isso, é calculado o valor de cada uma das respostas multiplicado o valor da resposta de acordo com a escala de Likert pela soma de vezes que este foi escolhido, dividido pela soma das vezes que o valor foi escolhido. A expressão abaixo mostra como foi calculado.

$$MP = \frac{(1\times S1)+(2\times S2)+(3\times S3)+(4\times S4)+(5\times S5)+(6\times S6)+(7\times S7)}{S1+S2+S3+S4+S5+S6+S7} \quad (1)$$

Todas as questões inseridas nas pesquisas têm a finalidade de observar um destes pontos: facilidade de uso, sentimento de segurança e aceitação do mecanismo.

Questionários que utilizam a escala de Likert permitem avaliar os itens separadamente ou em grupos, dependendo da forma como o questionário foi elaborado.

Como os questionários continham grupos de perguntas para avaliar pontos específicos, estas foram agrupadas para permitir uma melhor avaliação.

O Questionário 1 tem questões relacionadas com a percepção de segurança e com o conhecimento atual do participante sobre o sistema operacional do dispositivo, respondido pelos participantes antes de terem contato com o mecanismo. A Tabela 1 apresenta a frequência de cada resposta para cada afirmativa e a MP para cada uma das afirmativas.

Como dito anteriormente, a avaliação das perguntas ao questionário foi agrupada por assunto. As perguntas 2 e 3 estão relacionadas à utilidade do mecanismo para os participantes. O acúmulo das respostas está principalmente entre os valores 1 e 2 e a MP para a pergunta 1 é igual a 1,1 e para a pergunta 2 também é igual a 1,1.

Tabela 1 – Dispersão das respostas e Média Ponderada para o Questionário 1

Perguntas	Contagem de Respostas							MP
	1	2	3	4	5	6	7	
1 - Entendo como as aplicações instaladas em meu dispositivo utilizam suas permissões.	3	3	4	2	4	2	1	3.4
2 - Considero importante poder configurar as permissões das aplicações instaladas no dispositivo.	15	3	1	0	0	0	0	1.1
3 - Considero relevante ter garantia da forma como meus dados são utilizados.	19	1	0	0	0	0	0	1.1
4 - Confio na forma com que as aplicações utilizam os recursos do dispositivo.	3	1	0	8	3	2	2	3.9
5 - Sinto-me seguro quando utilizo meu dispositivo móvel.	1	1	5	6	3	2	1	3.8
6 - Considero que os dados em meu dispositivo estão seguros.	1	0	3	7	4	3	0	3.8
7 - Meu dispositivo oferece um ambiente para gerenciar as permissões requeridas pelas aplicações instaladas.	3	1	2	1	2	4	6	4.6
8 - Entendo como os mecanismos de segurança e privacidade de meu dispositivo funcionam.	1	2	4	6	3	2	1	3.9
9 - Sinto-me seguro com relação aos dados armazenados em meu dispositivo móvel.	1	1	2	4	7	3	1	4.3

Fonte: Elaborado pelo autor.

Dado o fato de que nenhum dos participantes tinha o conhecimento sobre a proposta ao responder o questionário, o resultado mostra o desejo dos participantes por uma aplicação de controle de segurança, o que é uma motivação para o desenvolvimento de mecanismos de controle de privacidade e segurança em dispositivos móveis.

Todas as respostas têm o objetivo de medir o sentimento de segurança atual dos participantes ao usar seus dispositivos móveis. Podemos notar nas respostas para essas perguntas um valor dispersivo, não mostrando uma tendência nas

respostas. Além disso, a MP para essas questões aponta para uma opinião neutra dos participantes. Isso pode ser justificado pelo baixo conhecimento dos problemas a que a permissão concedida a uma aplicação mal intencionada pode expor o proprietário do dispositivo.

A tendência do usuário de desejar um mecanismo para controlar a sua privacidade e a segurança apontou para um valor neutro, justificando o desenvolvimento do mecanismo e sua melhoria.

A Tabela 2 apresenta as respostas para o Questionário 2, cujo objetivo principal é avaliar a viabilidade do mecanismo, medindo facilidade de uso, aceitação e o sentimento de segurança quando o usuário utiliza um dispositivo com a proposta instalada. O objetivo dessa avaliação é a coleta de informações para melhorar a implementação do protótipo existente e também proporcionar referência para estudos futuros.

Tabela 2 – Dispersão das respostas e Média Ponderada para o Questionário 2

Perguntas	Contagem de respostas							MP
	1	2	3	4	5	6	7	
1 - De forma geral, estou satisfeito com a facilidade de uso do mecanismo MPP.	8	9	1	2	0	0	0	1.9
2 - Pude facilmente completar a tarefa com o mecanismo MPP.	9	7	4	0	0	0	0	1.8
3 - Fui capaz de efetuar um bloqueio de forma eficiente utilizando o mecanismo.	17	1	1	1	0	0	0	1.3
4 - Eu utilizaria este tipo de mecanismo em meu dispositivo.	15	2	3	0	0	0	0	1.4
5 - Foi fácil aprender como usar.	13	3	4	0	0	0	0	1.6
6 - Eu considero a proteção oferecida por este tipo de mecanismo válida.	16	2	1	0	0	0	0	1.5
7 - O sistema apresenta mensagens eficientes que me ajudam a utilizá-lo.	5	10	2	2	0	1	0	2.3
8 - O mecanismo MPP permite-me mudar facilmente meus bloqueios.	10	2	4	0	0	1	0	1.6
9 - Quando um bloqueio é realizado, a mensagem apresentada pelo mecanismo me ajuda claramente a entender a limitação na funcionalidade da aplicação.	7	9	1	1	0	1	1	2.3
10 - A forma com que os bloqueios são configurados é clara.	13	4	2	1	0	0	0	1.6
11 - A aplicação de configuração é agradável.	4	12	2	0	0	2	0	2.1
12 - O sistema tem todas as funcionalidades que eu desejo.	6	9	3	1	1	0	0	2.1
13 - No geral, estou satisfeito com o mecanismo.	11	7	2	0	0	0	0	1.7

14 - Considero importante poder configurar as permissões das aplicações instaladas no dispositivo.	19	1	0	0	0	0	0	1.1
15 - Considero relevante ter garantia da forma com que os dados são utilizados.	17	3	0	0	0	0	0	1.2
16 - Sinto-me seguro quando utilizo meu dispositivo móvel.	5	9	2	3	0	0	1	2.4
17 - Considero que os dados em meu dispositivo estão seguros.	4	6	5	4	0	0	1	2.7
18 - Confio na forma com que as aplicações utilizam os recursos do dispositivo.	3	7	2	3	3	2	0	3.1
19 – Sinto-me seguro com relação aos dados armazenados em meu dispositivo móvel.	4	6	4	3	2	0	1	3.0
20 - O mecanismo possibilitou que, a partir de agora, prestarei mais atenção na segurança de meus dispositivos móveis, tomando mais cuidado ao instalar aplicações.	16	2	0	1	1	0	0	1.5

Fonte: Elaborado pelo autor.

No Questionário 2, as perguntas de 2 a 4 foram inseridas para medir quanto os participantes julgam o protótipo fácil de usar. As respostas a estas questões estão acumuladas nos valores de aceitação. Valores entre 1 e 3 e com MP próxima a 1 mostram que o mecanismo é de fácil utilização.

Outro fato que reforça essa conclusão é que todos os participantes foram capazes de completar os três cenários de caso de uso sem ajuda. Esses fatos analisados em conjunto baseiam a conclusão de que a interface proposta foi fácil de aprender e pode ser usada como referência para implementações futuras.

As perguntas 7 e 8 foram inseridas na pesquisa para avaliar as mensagens e seus conteúdos. Essa avaliação é necessária pois é a forma como a proposta interage com o usuário do dispositivo quando um bloqueio de permissão é executado.

Mesmo que a MP para essa questão tenha apresentado um valor inferior a 3, e apontado para a aceitação das mensagens, esse é um ponto que precisa de atenção, pois todas as questões do questionário forneciam um campo adicional para o participante caso desejasse justificar sua resposta. Podemos notar que as questões relacionadas com mensagens tiveram o maior número de comentários.

Pode-se observar que os comentários no geral estavam relacionados às informações apresentadas e aos rótulos de permissão (por exemplo, informar que a regra pode ser alterada para conceder a permissão à aplicação).

Com base nessa observação, foi possível concluir que se trata de uma parte do mecanismo que necessita ser melhorada, pela importância dessa funcionalidade.

Com o intuito de medir a facilidade de uso e quanto os participantes apreciaram a interface proposta, as questões 9 e 13 foram inseridas na pesquisa. As respostas a essas perguntas mostram que a interface proposta pode ser utilizada como referência para futuras implementações, mas precisa de algumas melhorias.

As demais questões do Questionário 2 visavam a medir o sentimento de segurança dos participantes usando o dispositivo com o mecanismo instalado. A maioria das respostas foi acumulada em valores entre 1 e 3, e MP inferior a 3, o que indica que os participantes se sentem mais seguros quando o mecanismo está instalado.

Mas mesmo com esse retorno positivo, estudos futuros utilizando a proposta por um longo período certamente fornecerão informações sobre a segurança real e possibilitarão verificar se esse primeiro resultado foi influenciado pelo aspecto emocional, dado o fato de que o protótipo do mecanismo estava instalado no ambiente de teste e os participantes faziam parte de um estudo.

Como mencionado antes, cinco questões foram inseridas em ambos os questionários com a intenção de medir a alteração na percepção de segurança dos participantes ao utilizarem o dispositivo com e sem o mecanismo.

Tabela 3 – Respostas para as perguntas presentes nos dois questionários

Pergunta \ Valor		1	2	3	4	5	6	7
1	**S1**	15	3	1	0	0	0	0
	S2	19	1	0	0	0	0	0
2	**S1**	3	1	0	8	3	2	2
	S2	3	7	2	3	3	2	0
3	**S1**	1	1	5	6	3	2	1
	S2	5	9	2	3	0	0	1
4	**S1**	1	0	3	7	4	3	1
	S2	4	6	5	4	0	0	1
5	**S1**	1	1	2	4	7	3	1
	S2	4	6	4	3	2	0	1

Fonte: Elaborado pelo autor.

A **Erro! Fonte de referência não encontrada.** mostra a acumulação de cada resposta de acordo com a escala de Likert de 7 pontos para cada pergunta acima, no Questionário 1 (S1) e Questionário 2 (S2).

Estes valores foram utilizados para calcular a MP e traçar o gráfico apresentado na **Erro! Fonte de referência não encontrada.**, a fim de mostrar a mudança na sensação de segurança dos participantes utilizando o dispositivo, com e sem o mecanismo.

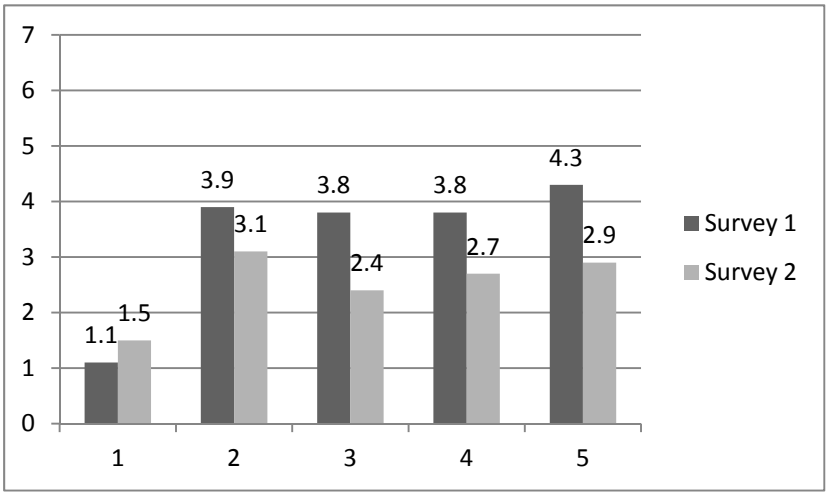
A equação utilizada para calcular a MP e gerar o gráfico foi a mesma utilizada para calcular a MP apresentada nas TabelasTabela 1 eTabela 2.

Na elaboração do gráfico, as questões foram separadas em dois grupos, um para a aceitação do protótipo implementado e outro para a percepção de segurança com e sem o mecanismo instalado no dispositivo.

A variável observada foi a alteração no sentimento de segurança nas resposta para as questões repetidas nos questionários 1 e 2. No Gráfico a primeira barra representa a MP para as respostas à questão no Questionário 1 e a segunda barra, a MP para as respostas no Questionário 2.

A avaliação estatística dos dados foi conduzida para a análise da importância da alteração na percepção de segurança dos participantes ao utilizarem o ambiente da avaliação de uso sem e com o mecanismo instalado.

Figura 29 – Comparativo entre as repostas apresentadas na Tabela 3



Fonte: Elaborado pelo autor.

A condução da avaliação estatística é especialmente importante quando se deseja avaliar os resultados. De uma forma sintética, pode-se dizer que a estatística

é um conjunto de técnicas apropriadas para recolher, classificar, apresentar e interpretar dados numéricos.

Utilizou-se o teste T de Student para amostras pareadas (BUSSAB; WILTON; MORETTIN; PEDRO, 2002) por se adequar à avaliação desejada: mesmos sujeitos em momentos distintos.

O teste T de Student – ou simplesmente teste T – é um teste de hipótese que utiliza conceitos estatísticos para rejeitar ou não uma hipótese nula, desde que o objeto de estudo siga uma distribuição. Teve sua origem em resposta à necessidade de aplicar métodos formais para estimar médias da população utilizando pequenas amostras.

Por se tratar de um teste de hipótese, o teste T gera uma hipótese nula e consequentemente uma hipótese alternativa. A hipótese nula é sempre sugerida com base em probabilidades estatísticas de uma determinada amostra ou população, sendo que a hipótese alternativa sempre confronta tais propriedades.

Assim, utilizando dados estatísticos, pode-se dizer que uma hipótese nula é verdadeira ou não, podendo, dessa forma, aceitá-la ou rejeitá-la. O teste T é muito utilizado quando se deseja avaliar a diferença entre as médias de dois grupos ou as médias de um mesmo grupo em momentos distintos.

Para aplicação do teste em amostras pequenas, é necessário assumir que a amostra é proveniente de uma distribuição normal e utilizar um grau de liberdade igual a n-1 onde n é o número de participantes ou amostras coletadas.

Também deve-se estabelecer o intervalo de confiança, que define qual é a probabilidade de que o intervalo estimado contenha o parâmetro populacional.

Para determinar se a alteração nas medias é significativa, foi utilizado o conceito de significância estatística. Esse conceito nos ajuda a determinar quão verdadeira é a hipótese que se está testando.

Em geral, a significância é representada através do valor P, e a definição de seus limites deve ser feita antes do início do estudo. O Quadro 5 apresenta os valores utilizados na avaliação.

Quadro 5 – Significância estatística

Valor P	Significado
>0,05	Não significante
0,01 a 0,05	Significante
0,001 a 0,01	Muito significante
<0,001	Extremamente significante

Fonte: Elaborado pelo autor.

Assim, para a execução do teste, foi considerado um intervalo de confiança de 95%, tendo assim uma significância de 0.05. Dado que a amostra foi composta por 20 participantes e o grau de liberdade para o teste T é n-1, temos um grau de liberdade de 19.

Como mencionado, queremos verificar o quão significante foi a alteração na MP das cinco questões do Questionário 1 para o Questionário 2, considerando a escala de Likert apresentada no Quadro 4, em que valores próximos do valor 1 representam a total concordância com a questão. Assim, podemos concluir essas hipóteses:

H0: não há diferença entre as MP nos questionários 1 e 2 (MP1 - MP2 = 0);
H1: há diferença entre as MP nos questionários 1 e 2 (MP2 - MP1 <> 0).

Isso significa que:
H0: MPP não faz diferença;
H1: MPP faz diferença.

Como estamos considerando valores de médias diferentes da hipótese nula, foi realizado um teste bilateral em que os pontos críticos são -T/2 e T/2 assumindo o grau de liberdade apresentado.

Durante a avaliação dos resultados, também foi utilizado o conceito estatístico de valor P, que é o nome que se dá à probabilidade de se observar um resultado tão ou mais extremo que o da amostra inicial, supondo que a hipótese nula seja verdadeira.

Assim sendo, durante a avaliação, considerando o valor de significância de 0,05; entenderemos que houve uma mudança significativa na percepção de

segurança pelos participantes e na aceitação do mecanismo, caso o valor P seja menor que 0.05. O teste T para cada declaração será apresentado a seguir.

O Quadro 6 apresenta as médias, o desvio padrão e a variância calculados para a primeira questão. Os valores intermediários utilizados nos cálculos foram t = 1,4530, distribuição tn-1 = 19, erro padrão da diferença = 0,138.

O intervalo de confiança, definido como sendo a média do Questionário 1 subtraída a média do Questionário 2, é igual a 0,20, considerando o intervalo de confiança de 95% dessa diferença: a partir de -0,09 a 0,49.

Quadro 6 – Teste T - questão 1

	Questionário 1	**Questionário 2**
Média	1.25	1.05
Desvio Padrão	0.55	0.22
Variância	0.12	0.05
Nº participantes	20	20

Fonte: Elaborado pelo autor.

Com base nos cálculos, foi obtido um valor P bicaudal equivalente a 0,1625. Pelo critério definido no Quadro 5, essa diferença é considerada estatisticamente não significativa.

Essa questão, como foi mencionado, foi inserida com o intuito de mensurar o desejo dos participantes por uma ferramenta para o controle de privacidade e segurança e de verificar se o sentimento dos participantes se alterava após o contato com o protótipo.

Como apresentado na avaliação do Questionário 1, as respostas a essa questão já indicavam o desejo dos participantes por ferramentas para o controle da privacidade e segurança, sendo um bom motivador para o desenvolvimento de propostas como a apresentada neste trabalho.

A avaliação aqui reforça essa conclusão, pois, mesmo após o contato com o protótipo e a ciência das configurações necessárias para o mecanismo realizar bem seu propósito, não houve mudança significativa no desejo dos participantes.

O Quadro 7 apresenta as médias, o desvio padrão e a variância calculada para a segunda questão. Os valores intermediários utilizados nos cálculos foram t = 2,2380, distribuição tn-1 = 19, erro padrão da diferença = 0,492.

O intervalo de confiança, definido como sendo a média do Questionário 1 subtraída a média do Questionário 2, é igual a 1,10, considerando o intervalo de confiança de 95% desta diferença: de 0,07-2,13.

Quadro 7 – Teste T – questão 2

	Questionário 1	**Questionário 2**
Média	4.15	3.05
Desvio Padrão	1.79	1.61
Variância	0.4	0.36
Nº participantes	20	20

Fonte: Elaborado pelo autor.

Com base nos cálculos, foi obtido um valor P bicaudal equivalente a 0,0374. Pelo critério definido, essa diferença é considerada como sendo estatisticamente significativa.

Essa afirmativa tinha como objetivo avaliar a confiança nas aplicações instaladas no dispositivo. A mudança significativa para essa afirmativa é uma característica importante inserida pela proposta, pois mostra que o mecanismo fez com que os participantes tivessem mais atenção no que cada aplicação está realizando.

O Quadro 8 apresenta as médias, o desvio padrão e a variância calculada para a terceira questão. Os valores intermediários utilizados nos cálculos foram t = 3,5136, distribuição tn-1 = 19, Erro Padrão da Diferença = 0455.

O intervalo de confiança **é** definido como sendo a média do Questionário 1 subtraída a média do Questionário 2 e é igual a 1,60, considerando o intervalo de confiança de 95% desta diferença: de 0,65-2,55.

Quadro 8 – Teste T – questão 3

	Questionário 1	**Questionário 2**
Média	4.4	2.7
Desvio Padrão	1.35	1.45
Variância	0.3	0.33
Nº participantes	20	20

Fonte: Elaborado pelo autor.

Com base nos cálculos, foi obtido um valor P bicaudal equivalente a 0,0023. Pelo critério definido, essa diferença é considerada estatisticamente muito significativa.

A alteração notada nessa avaliação é muito importante para o estudo aqui realizado, pois demonstra que a proposta proporcionou um maior sentimento de segurança nos participantes ao utilizarem o dispositivo com o mecanismo proposto instalado.

O Quadro 9 apresenta as médias, o desvio padrão e a variância calculada para a quarta questão. Os valores intermediários utilizados nos cálculos foram t = 5,3618, distribuição tn-1 = 19, erro padrão da diferença = 0,317. O intervalo de confiança foi igual a 1,70, considerando intervalo de confiança de 95% dessa diferença: de 1,04-2,36.

Quadro 9 – Teste T - questão 4

	Questionário 1	**Questionário 2**
Média	4.0	2.4
Desvio Padrão	1.41	1.47
Variância	0.32	0.33
Nº participantes	20	20

Fonte: Elaborado pelo autor.

Com base nos cálculos foi obtido um valor P bicaudal inferior a 0,0001. Pelo critério definido, essa diferença é considerada na estatística como sendo extremamente significativa.

Essa avaliação também nos remete a um ponto favorável inserido pela proposta, pois fez com que os participantes se preocupassem mais com as informações armazenadas no dispositivo.

O

Quadro 10 apresenta as médias, o desvio padrão e a variância calculados para a quinta questão. Os valores intermediários utilizados nos cálculos foram t = 3,7004, distribuição tn-1 = 19, erro padrão da diferença = 0,459.

O intervalo de confiança foi igual a 1,70, considerado o intervalo de confiança de 95% dessa diferença: de 0,74-2,66.

Quadro 10 – Teste T - questão 5

	Questionário 1	Questionário 2
Média	4.55	2.85
Desvio Padrão	1.47	1.6
Variância	0.33	0.36
Nº participantes	20	20

Fonte: Elaborado pelo autor.

Com base nos cálculos foi obtido um valor P bicaudal equivale a 0,0015. Pelo critério definido, essa diferença é considerada estatisticamente muito significativa, indicando que a utilização da proposta, o conhecimento de seus propósitos e a verificação de suas funcionalidades proporcionaram um maior sentimento de segurança dos participantes com relação a seus dados

6.1.3 Considerações finais

A Avaliação de Uso demonstrou que o modelo aqui proposto é válido e de fácil utilização. Essa conclusão foi baseada na análise estatística bem como nas observações de campo como, por exemplo, o fato de todos os participantes executarem os cenários de caso de uso sem ajuda.

Também é relevante mencionar que os participantes julgaram importante a implementação de mecanismos de gerenciamento da segurança, como apresentado na avaliação do Questionário 2 e na avaliação da primeira afirmativa inserida em ambos os questionários.

Uma das conclusões mais importantes obtidas com a avaliação de uso é que os usuários de dispositivos móveis podem ter um falso sentimento de segurança.

Isso ocorre porque, apesar de terem conhecimento de que as aplicações precisam de permissões para serem executadas, concedidas no momento de instalação da aplicação, os usuários não sabem para qual funcionalidade é concedida a permissão, nem o riscos a que podem estar expostos se a permissão for utilizada de forma maliciosa.

6.2 **Avaliação de desempenho**

Com o objetivo de avaliar o desempenho do mecanismo desenvolvido, foram realizados testes para medir o atraso ocasionado por ele. Para tanto, foram utilizadas as seguintes aplicações: o browser de acesso à *Internet* e o aplicativo de envio de SMS. Também foi avaliado o atraso nas verificações da plataforma durante a inicialização.

Foram criadas regras de bloqueio personalizadas, por meio da interface desenvolvida, como parte da aplicação de prova de conceito e computado o tempo consumido para a realização da permissão sem e com um bloqueio definido. Porém, nos dois casos existiam regras a serem consultadas no mecanismo.

O valor base assumido como referência é zero, pois o tempo gasto pelo mecanismo já existente na plataforma foi desconsiderado, uma vez que deve ser realizado de qualquer forma. Assim, os resultados aqui apresentados dizem respeito apenas aos atrasos provocados pelo protótipo implementado.

Estudos com relação ao consumo de memória e processamento pelo mecanismo não foram realizados. Contudo, os mesmos serão necessários na hipótese de a implementação vir a ser instalada nos dispositivos dos usuários.

Nos testes de desempenho, foram realizadas 10 medições do tempo consumido para a realização do bloqueio da permissão solicitada, no caso do Browser. O tempo consumido para conceder o acesso configurado no mecanismo também foi medido.

Na Tabela 4, é apresentado o tempo necessário para a realização de um bloqueio em 10 solicitações de acesso à *Internet* pelo aplicativo de browser. O tempo gasto foi apresentado na tabela em milissegundos, tendo sido computado do início da verificação até o seu final, quando o bloqueio já foi realizado. Também foi apresentada a média do tempo consumido.

O maior tempo gasto foi de 105 milissegundos, que é absolutamente aceitável em face da funcionalidade do mecanismo. Esse valor também não é constante e foi notado apenas uma vez, tendo o tempo médio de execução sido estabelecido em 43,8 milissegundos, concluindo-se que o mecanismo provoca um atraso mínimo.

Tabela 4 – Tempo consumido para bloquear o acesso à Internet no Browser

Início	Fim	Gasto
1360897625098,00	1360897625142,00	44
1360897653784,00	1360897653815,00	31
1360897675121,00	1360897675174,00	53
1360897695188,00	1360897695227,00	39
1360897712280,00	1360897712314,00	34
1360897742213,00	1360897742246,00	33
1360897770837,00	1360897770871,00	34
1360897795756,00	1360897795861,00	105
1360897832835,00	1360897832865,00	30
1360897854309,00	1360897854344,00	35
	Média	**43,8**

Fonte: Elaborado pelo autor.

A Tabela 5 apresenta o tempo de início, fim e o tempo médio gasto para a liberação de uma permissão de acesso. Os tempos apresentados são relativos ao momento em que o mecanismo faz o desvio da plataforma para efetuar a verificação de permissão. Cada liberação de acesso à *Internet* não superou 45 milissegundos e a média foi de 35,3 milissegundos, um valor aceitável.

Tabela 5 – Tempo consumido para liberar acesso à Internet no Browser

Início	Fim	Gasto
1360898043683,00	1360898043728,00	45
1360898071376,00	1360898071408,00	32
1360898092633,00	1360898092664,00	31
1360898115756,00	1360898115787,00	31
1360898146981,00	1360898147024,00	43
1360898164538,00	1360898164570,00	32
1360898181158,00	1360898181198,00	40
1360898204614,00	1360898204645,00	31
1360898225364,00	1360898225395,00	31
1360898243506,00	1360898243543,00	37
	Média	**35,3**

Fonte: Elaborado pelo autor.

Para o aplicativo de SMS, só serão apresentados os tempos gastos para a realização do bloqueio. A Tabela 5 apresenta o tempo gasto para a realização do bloqueio da permissão de envio de SMS. Para a realização desse teste, foi utilizado o aplicativo de SMS disponível na plataforma Android.

Foi configurado o bloqueio da permissão de envio de SMS no mecanismo e em seguida foi realizada a tentativa de envio de 10 mensagens SMS. O início, fim e o tempo consumido para a realização de cada bloqueio são apresentados na Tabela 6.

Os resultados obtidos pelos testes de atraso apresentados neste capítulo indicam que o atraso inserido pelo mecanismo é mínimo e totalmente aceitável, dada a funcionalidade que o mesmo fornece.

Tabela 6 – Tempo consumido para bloquear envio SMS

Início	Fim	Gasto
1360897163069,00	1360897163085,00	16
1360897256231,00	1360897256246,00	15
1360897201750,00	1360897201768,00	18
1360897229566,00	1360897229582,00	16
1360897291310,00	1360897291339,00	29
1360897315546,00	1360897315576,00	30
1360897346133,00	1360897346150,00	17
1360897379717,00	1360897379734,00	17
1360897415477,00	1360897415493,00	16
1360897446757,00	1360897446774,00	17
	Média	**19,1**

Fonte: Elaborado pelo autor.

Estudos futuros no sentido de avaliar um mecanismo que não exija a mudança no núcleo do sistema operacional ou que utilize um modelo de armazenamento de regras mais veloz podem ajudar na diminuição dos tempos aqui apresentados.

Capítulo 7

CAPÍTULO 7 - **CONCLUSÕES E TRABALHOS FUTUROS**

Este capítulo apresenta os as conclusões obtidas durante o desenvolvimento da proposta apresentada neste trabalho, bem como sugestões de trabalhos futuros.

O mecanismo aqui apresentado não só proporciona maior segurança dos dispositivos, mas também possibilita a seus usuários o gerenciamento da forma com os recursos de seus dispositivos são utilizados pelas aplicações instaladas.

Essa é uma conclusão baseada na avaliação estatística da comparação entre o sentimento de segurança com e sem a proposta instalada no dispositivo.

A proposta se mostrou útil para a definição de regras de segurança personalizadas por usuários ou empresas. No caso de implementações para o meio corporativo, estas podem ser utilizadas para o controle de privacidade e segurança de dispositivos móveis usados pelos funcionários para acessar recursos da empresa.

Dessa forma, a empresa terá o controle sobre as regras de segurança dos dispositivos, diminuindo, assim, o risco de acessos maliciosos por aplicações instaladas.

Entre as principais conclusões obtidas com os experimentos realizados, podemos apontar o fato de que o mecanismo mostrou-se útil e de fácil utilização, o que é comprovado pelas respostas ao Questionário 2, bem como pelo fato de todos os participantes terem realizado os três cenários de estudo de caso durante a Avaliação de Uso de forma independente.

Foi possível também concluir que o mecanismo é útil e que os participantes desejam ter o controle de sua privacidade e segurança. Tal conclusão pode ser

observada por meio das respostas fornecidas aos dois questionários e da avaliação estatística realizada na questão referente à utilização do mecanismo.

Tal avaliação mostrou que os participantes, mesmo antes de conhecer o mecanismo, tinham interesse em um mecanismo de controle de privacidade e segurança e o interesse por ferramentas dessa natureza não mudou após o contato com o mecanismo, o que indica que, mesmo não sendo uma versão final, o protótipo implementado demonstrou ser uma boa referência para implementações futuras.

Para o protótipo implementado, um dos principais pontos de melhoria a ser explorados, de acordo com os participantes, refere-se à notificação de bloqueios e à disponibilização de funcionalidades que auxiliem na utilização do mecanismo, como a criação de sinônimos para as permissões, recurso autocompletar e ajuda. Essa informação é importante não só para o trabalho aqui proposto mas também para estudos futuros, servindo de referência para implementações mais amigáveis.

Com relação à privacidade e segurança, o questionário inicial indica que os participantes possuem uma certa indecisão com relação à segurança atual quando utilizando dispositivos móveis.

Tal fato pode ser justificado pelo fato de os participantes, em sua maioria, possuírem o conhecimento de que as aplicações solicitam permissões no momento da instalação. Porém, os mesmos não têm conhecimento da forma como estas são utilizadas ou os riscos que a má utilização pode ocasionar, bem como acreditam que o sistema operacional em seus dispositivo já realiza tal controle.

A avaliação estatística realizada com as questões inseridas em ambos os questionários relativas ao sentimento de segurança quando utilizando o dispositivo com e sem o mecanismo aqui proposto mostrou que os participantes tinham um sentimento maior de segurança quando usaram o dispositivo com o mecanismo instalado. Esse é um fato que ajuda a validar o mecanismo proposto e a realização de estudos futuros com o mesmo foco do mecanismo aqui proposto.

Porém, essas observações merecem um estudo mais aprofundado com o intuito de validar tais resultados, pois os resultados iniciais podem ter sido influenciados pelo fato de os participantes estarem fazendo parte de um estudo e saberem o objetivo do mecanismo que estavam testando.

Um estudo em que os participantes respondessem ao questionário inicial e em seguida utilizassem o mecanismo por um período maior de tempo (dias, semanas ou meses) colaboraria com a validação desses resultados e a avaliação da

perspectiva de outras variáveis, como o uso no dia a dia e o impacto do mecanismo nas tarefas diárias dos participantes.

Com relação ao desempenho, o mecanismo provou inserir valores aceitáveis de atraso durante a execução de sua funcionalidade, sendo o maior tempo observado de 0,1 segundo, indicando que o impacto para o usuário final é mínimo e provando a viabilidade do mecanismo proposto neste trabalho. Estudos futuros com o intuito de melhorar o desempenho do mecanismo podem aumentar a satisfação dos usuários com a implementação do mecanismo sem a necessidade de alterar o núcleo do sistema operacional, auxiliando seu desempenho e diminuindo os atrasos gerados às aplicações

Dentre as contribuições deste trabalho destacam-se a apresentação da proposta de um ferramenta de controle de privacidade personalizada, a conscientização do usuário sobre o que cada aplicação está fazendo, o incentivo do desejo dos participantes por esse tipo de mecanismo, bem como ser uma referência e fornecer dados para trabalhos futuros

Como proposta de trabalhos futuros indicam-se uma implementação que não necessite a alteração do núcleo do sistema operacional, possibilitando assim a utilização em dispositivos diversos, e a implementação da proposta em outra plataforma a fim de comparar com o protótipo existente.

Outra pesquisa que pode contribuir com a evolução do trabalho aqui apresentado é um estudo em que os participantes utilizariam a proposta por um período maior de tempo: semanas ou meses.

Entre os avanços proporcionados pelo mecanismo desenvolvido nesta pesquisa, podemos citar a criação de uma ferramenta não existente; o aumento da consciência dos usuário sobre as aplicações executadas em seus dispositivos, bem como um novo ponto de vista sobre a disponibilização ou não de ferramentas para gerenciamento de permissões em dispositivos móveis.

ANDROID. Android, the world's most popular mobile platform. [S.l.]: Developers. Welcome.. Disponível em: <http://developer.android.com/guide/basics/what-is-android.html>. Acesso em: 21 abr.2012a.

ANDROID. Application Fundamentals. [S.l.]: Developers. API Guides. Disponível em: <http://developer.android.com/guide/topics/fundamentals.html>. Acesso em: 17 abr.2012b.

ANDROID. Developers. [S.l..]: Device Administration. 2012a. Disponível em: <http://developer.android.com/guide/topics/admin/device-admin.html>. Acesso em: 7 jan.2012c.

ANDROID. Devices. Technical Information. Disponível em: <http://source.android.com/tech/security/index.html#the-android-permission-model-accessing-protected-apis>. Acesso em 20 jan. 2012d.

ANDROID. Reference [S.l.]: Developers. Android APIs. Disponível em: <http://developer.android.com/reference/android/content/Intent.html>. Acesso em: 15 abr.2012e.

ANDROID. Security Tips. [S.l.]: Developers. Training. Disponível em: <http://developer.android.com/guide/topics/security/security.html>. Acesso em: 17 abr.2012f.

ANDROID. The Android Source Code. [S.l;]: Source, 2012b. Disponível em: <http://source.android.com/source/index.html>. Acesso em: 21 abr.2012g.

APPLE. iOS developer Library: Security. Disponível em <http://images.apple.com/ipad/business/docs/iOS_Security_May12.pdf>. Acesso em: 1 mar.2013.

BERESFORD, Alastair R.; RICE, Andrew; SKEHIN, Nicholas; SOHAN, Ripduman. 2011. MockDroid: trading privacy for application functionality on smartphones. In**:** **Proceedings of the 12th Workshop on Mobile Computing Systems and Applications** (HotMobile '11). ACM, New York, NY, USA, 49-54. Disponível em: <http://doi.acm.org/10.1145/2184489.2184500>. Acesso em: 2 mar.2012.

BUSSAB, Wilton de O.; MORETTIN, Pedro A. **Estatística Básica.** 5.ed. São Paulo: Saraiva, 2002. . ISBN 85-02-03497-9. Acesso em: 16 jan.2013

CHANDRAMOHAN, M.; TAN, H.B.K. **Detection of Mobile Malware in the Wild.** IEEE Computer Society, 2010. Disponível em: <http://www.computer.org/portal/web/csdl/doi/10.1109/MC.2012.36>. Acesso em: 15 mar.2012.

CONTI, M.; NGUYEN, V.; CRISPO, B. CRePE: Context-related policy enforcement for Android. **Information Security**, 2011. p. 331-345. Disponível em: <http://www.springerlink.com/content/574882rn4t65364m/>. Acesso em: 15 mar.2012.

DELAC, G.; SILIC, M.; KROLO, J. Emerging security threats for mobile platforms. MIPRO, 2011 **Proceedings of the 34th International Convention**, p. 1468-1473p. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5967292&isnumber=5967009>. Acesso em: 15 mar.2012.

DENNIS, J.B.; VAN HORN, E.C. Programming Semantics for Multiprogrammed Computations. **Comm. ACM**, v. 9, March 1966. p. 143-154. Disponível em: <http://dl.acm.org/citation.cfm?id=357993>. Acesso em: 15 mar.2012.

ENCK, W.; ONGTANG, M.; MCDANIEL, P. Understanding Android Security. **Security & Privacy**, IEEE, v. 7, n.1, Jan.Feb. 2009. p.50-57. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4768655&isnumber=4768640>. Acesso em: 21 fev.2012.

FIDEL, R. **Quality Methods in Information Retrieval Research**. 1993. Disponível em: <http://faculty.washington.edu/fidelr/RayaPubs/QualitativeMethodsinInformationRetrievalResearch.pdf >. Acesso em: 7 dez.2012.

GOOGLE. Android Home Page, 2009. Disponível em: <http://www.android.com>. Acesso em: 21 abr.2012.

HOLZINGER, A. Usability Engineering Methods for Software Developers. . **Communications of the ACM**. v. 48, n. 1, 2005. p. 71-74. Acesso em: 22 abr.2012.

HUGHES, E. A **Cypherpunk's Manifesto**. 1993. Disponível em: <http://www.activism.net/cypherpunk/manifesto.html>. Acesso em 15 mai.2013.

LIU, Z.; NAM, C.S.; SHIN, D.R. UAMDroid: A user authority manager model for the Android platform. Advanced Communication Technology (ICACT), 2011. **13th International Conference**, 2011. p. 1146-1150. Disponível em: <http://ieeexplore.ieee.org/xpl/ freeabs_all.jsp? arnumber=5746009>. Acesso em: 23 jan.2012.

MATTAR, Fauze Najib. **Pesquisa de marketing**. Edição Compacta. 3.ed. São Paulo: Atlas, 2001. Acesso em: 21 jan.2013.

MOWEN, John C.; MINOR, Michael S**. Comportamento do consumidor**. São Paulo: Prentice Hall, 2003. Acesso em: 21 jan.2013.

NAUMAN, M.; KHAN; S; ZHANG, X. Apex: Extending Android permission model and enforcement with user-defined runtime constraints. **Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security**, 2010. p 328-332. Disponível em: <http://dl.acm.org/citation.cfm?id=1755732>. Acesso em: 8 dez.2011.

NOKIA. Symbian OS Platform Security. 2012. Disponível em: <http://www.developer.nokia.com/Community/Wiki/Symbian_OS_Platform_Security/02._Platform_Security_Concepts>. Acesso em: 7 nov.2011.

ONGTANG, M.; MCLAUGHLIN; S.; ENCK, W.; MCDANIEL, P. Semantically rich application-centric security in android. **Computer Security Applications Conference,** 2009. ACSAC'09. Annual, p. 340-349. Disponível em: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5380692>. Acesso em: 27 nov.2011.

RAINER KUHLEN. Informationsethik. Umgang mit Wissen und Information in elektronischen Räumen. Universitätsverlag Konstanz 2004. Acesso em: 15 nov.2011.
RAMOS, P. . El 80% de los smartphones no están protegidos contra amenazas. [S.l.]: ESET. Blog de laboratório. 14 jun.2011. Disponível em: <http://blogs.eset-la.com/laboratorio/2011/06/14/smartphones-protegidos-amenazas/>. Acesso em: 21 fev.2012

SALTZER, J.; SCHROEDER, M. **The Protection of Information in Computer Systems**, 1975. Disponível em: <http://www.cs.virginia.edu/~evans/cs551/saltzer/>. Acesso em: Acesso em: 13 nov.2011.

SMITH, R.E. A Contemporary Look at Saltzer and Schroeder's 1975 Design Principles. Security & Privacy, IEEE , v.10, n. 6, p.20-25, Nov.-Dec. 2012, Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6226346&isnumber=6375711>. Acesso em: 1 mar.2012.

SPIEGEL, M. **Estatística**: Resumo da Teoria. Rio de Janeiro: McGraw-Hill, 1971.
TRAUTH, E. M; O'CONNOR, B. A study of the interaction between information, technology and society: an illustration of combined qualitative research methods. In: **Information Systems Research**: Contemporary Approaches & Emergent Traditions. Amsterdam, 1991, p.131-144. Acesso em: 24 jan.2013.

YIN, R. K. **Estudo de Caso**: Planejamento e método. 3. ed. Porto Alegre: Bookman, 2005. Acesso em: 27 jan.2013.
ZEMIN LIU; CHOON-SUNG NAM; DONG-RYEOL SHIN. UAMDroid: A user authority manager model for the Android platform. Advanced Communication Technology (ICACT), 2011 **13th International Conference**, p.1146-1150, 13-16, Feb. 2011. Disponível em:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5746009&isnumber=5745722>. Acesso em: 15 mar.2012.

APÊNDICE I - CENÁRIOS DE USO

Instruções de uso do simulador

1- Botão que retorna para a tela inicial do simulador.

2- Botão que retorna para a última tela e para esconder o teclado.

3- Ao clicar em uma caixa de texto o teclado se abre automaticamente.

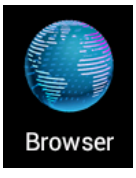
4 - Ícone para criar um novo SMS.

Cenário 1

O objetivo deste cenário é a realização do acesso à *Internet* por meio do aplicativo de browser e em seguida configurar o bloqueio da permissão.

1 - No simulador, vá ao ícone para abrir a lista de aplicações.

2- Abra o ícone do browser e tente acessar algum site.

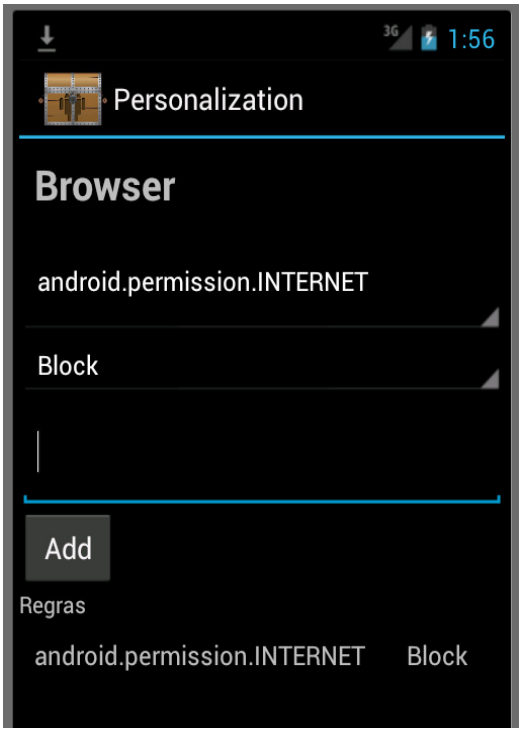


3 - Utilizando o botão 2 retorne até a lista de aplicações.

4 - Em seguida selecione a aplicação de gerenciamento do MPP.

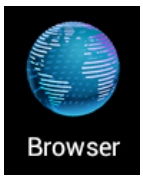
5- Quando o MPP abrir selecione:

6- Na tela de personalização selecione a primeira opção "android.permission.INTERNET" e "Block", como na figura abaixo, em seguida pressione o botão Add.



7 - Utilizando o botão 2 retorne até a lista de aplicações.

8- Abra o ícone do browser e tente acessar algum site.



Cenário 2

O objetivo deste cenário é a realização do bloqueio de envio de SMS.

1 - No simulador, vá ao ícone para abrir a lista de aplicações.

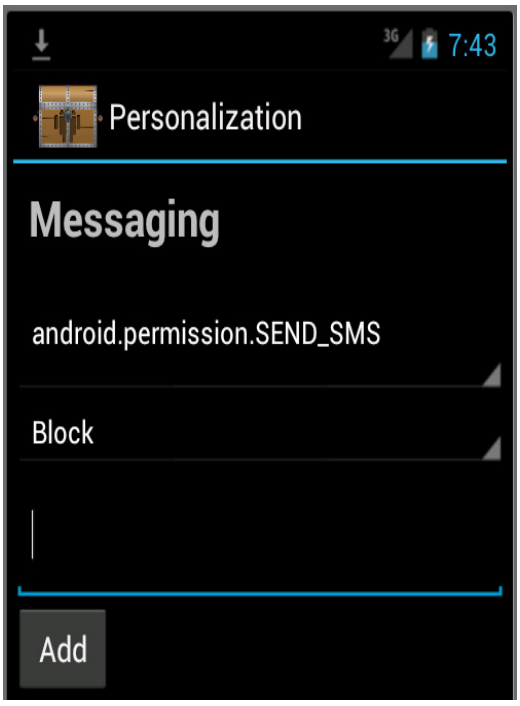
2 - Abra o ícone do aplicativo de mensagens e tente enviar uma SMS.

O número e a mensagem podem ser aleatórios.

3 - Utilizando o botão 2, retorne até a lista de aplicações.

4 - Em seguida selecione a aplicação de gerenciamento do MPP.

5 - Quando o MPP abrir, selecione:



6- Na tela de personalização selecione a primeira opção "android.permission.SEND_SMS" e "Block", como na figura, em seguida pressione o botão Add.

7 - Utilizando o botão 2, retorne até a lista de aplicações.

8 – Abra o ícone do aplicativo de mensagens e tente enviar uma SMS.

O número e a mensagem podem ser aleatórios.

Cenário 3

O objetivo deste cenário é apresentar um bloqueio que pode ocasionar na não execução da aplicação.

1- No simulador, vá ao ícone para abrir a lista de aplicações.

2 - Abra o ícone do aplicativo de mensagens e tente enviar uma mensagem.

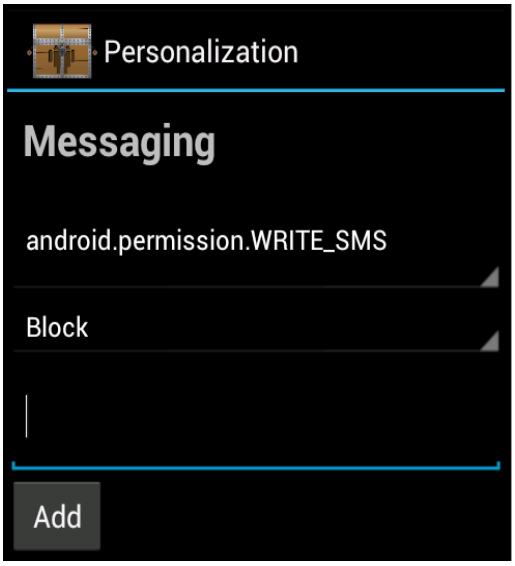
O número e a mensagem podem ser aleatórios.

3 - Utilizando o botão 2 retorne até a lista de aplicações.

4 - Em seguida selecione a aplicação de gerenciamento do MPP.

5 - Quando o MPP, abrir selecione:

6 - Na tela de personalização selecione a primeira opção "android.permission.WRITE_SMS" e "Block", como na figura abaixo, em seguida pressione o botão Add.



7 - Utilizando o botão 2, retorne até a lista de aplicações.

8 - Abra o ícone do aplicativo de mensagens e tente enviar uma SMS. O número e a mensagem podem ser aleatórios.

APÊNDICE II - APRESENTAÇÃO

Estudo de caso

MPP - Mecanismo de personalização de privacidade

Você está sendo convidado para participar da pesquisa de estudo de caso do projeto de mestrado “Satisfação com o Uso da Ferramenta MPP". Você foi selecionado por sua formação profissional, sua familiaridade na utilização de dispositivos móveis e a plataforma na qual o protótipo foi implementado e facilidade de acesso ao local onde serão realizados os experimentos. No entanto sua participação não é obrigatória.

Este estudo de caso será composto por três etapas

1. **Reposta a questionário inicial**
   Esta etapa é importante para coletar informações sobre a segurança antes da apresentação do mecanismo.
2. **Apresentação e utilização do mecanismo**
   Nesta etapa será apresentada uma breve descrição do mecanismo e em seguida os participantes irão executar os casos de uso.
3. **Questionário final**
   Nesta etapa será solicitado aos participantes que respondam o questionário com base nas observações feitas durante o caso de uso.

Para responder ao questionário com base nas observações durante o caso de uso, o participante irá escolher um valor entre 1 e 7 onde:

1. Concordo Fortemente
2. Concordo
3. Concordo parcialmente
4. Indiferente
5. Discordo parcialmente
6. Discordo
7. Discordo Fortemente

Obrigado,
Leonardo Leite de Melo e Prof. Dr. Sérgio Donizette Zorzo

APÊNDICE III - QUESTIONÁRIO 1

Questionário 1

1 – Entendo como as aplicações instaladas em meu dispositivo utilizam suas permissões.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

2 –Considero importante poder configurar as permissões da aplicações instaladas no dispositivo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

3 – Considero relevante ter garantia da forma como meus dados são utilizados.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

4 – Confio na forma com que as aplicações utilizam os recursos do dispositivo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

5 - Sinto-me seguro quando utilizo meu dispositivo móvel.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

6 - Considero que os dados em meu dispositivo estão seguros.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

7 – Meu dispositivo oferece um ambiente para gerenciar as permissões requeridas pelas aplicações instaladas.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

8 – Entendo como os mecanismos de segurança e privacidade de meu dispositivo funcionam.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

9 - Sinto–me seguro com relação aos dados armazenados em meu dispositivo móvel.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

Seguem abaixo as testemunhas pela validade e veracidade das respostas dadas acima.

São Carlos, ___ de _____ de 2012.

_____	_____
Leonardo Leite Melo	Participante da Pesquisa

APÊNDICE IV - QUESTIONÁRIO 2

Questionário 2

1. De forma geral, estou satisfeito com a facilidade de uso do mecanismo MPP.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

2. Pude facilmente completar a tarefa com o mecanismo MPP.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

3. Fui capaz de efetuar um bloqueio de forma eficiente utilizando o mecanismo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

4. Eu utilizaria este tipo de mecanismo em meu dispositivo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

5. Foi fácil aprender como usar.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

6. Eu considero a proteção oferecida por este tipo de mecanismo válida.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

7. O sistema apresenta mensagens eficientes que me ajudam a utilizá-lo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

8. O mecanismo MPP permite-me mudar facilmente meus bloqueios.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

9. Quando um bloqueio é realizado, a mensagem apresentada pelo mecanismo me ajuda claramente a entender a limitação na funcionalidade da aplicação.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

10. A forma com que os bloqueios são configurados é clara.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

11. A aplicação de configuração é agradável.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

12. O sistema tem todas as funcionalidades que eu desejo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

13. No geral estou satisfeito com o mecanismo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

14 – Considero importante poder configurar as permissões da aplicações instaladas no dispositivo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

15 – Considero relevante ter garantia da forma com que os dados são utilizados.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

16 - Sinto me seguro quando utilizo meu dispositivo móvel.

Concordo Fortemente 1 2 3 4 5 6 7 **Discordo Fortemente**

Comentários:

17 - Considero que os dados em meu dispositivo estão seguros.

Concordo Fortemente 1 2 3 4 5 6 7 **Discordo Fortemente**

Comentários:

18 – Confio na forma com que as aplicações utilizam os recursos do dispositivo.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

19 - Sinto-me seguro com relação aos dados armazenados em meu dispositivo móvel.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

21 – O mecanismo possibilitou que, a partir de agora, prestarei mais atenção na segurança de meus dispositivos móveis, tomando mais cuidado ao instalar aplicações.

Concordo Fortemente 1 2 3 4 5 6 7 Discordo Fortemente

Comentários:

Seguem abaixo as testemunhas pela validade e veracidade das respostas dadas acima.

São Carlos, ___ de _____ de 2012.

_____ _____

Leonardo Leite Melo Participante da Pesquisa

APÊNDICE IV - SLIDES DE APRESENTAÇÃO DO MECANISMO

Motivação

- Tecnologias presentes
- Quantidade de dados
- Como as aplicações usam as permissões

Objetivo

- Transferir o controle de permissões ao usuário
- Permissões dinâmicas e personalizadas
- Aumento da privacidade e segurança do usuário

MPP

- Mediação de todas requisições
- Verificação antes da plataforma
- Regras definidas pelo usuário
- Verificação dinâmica

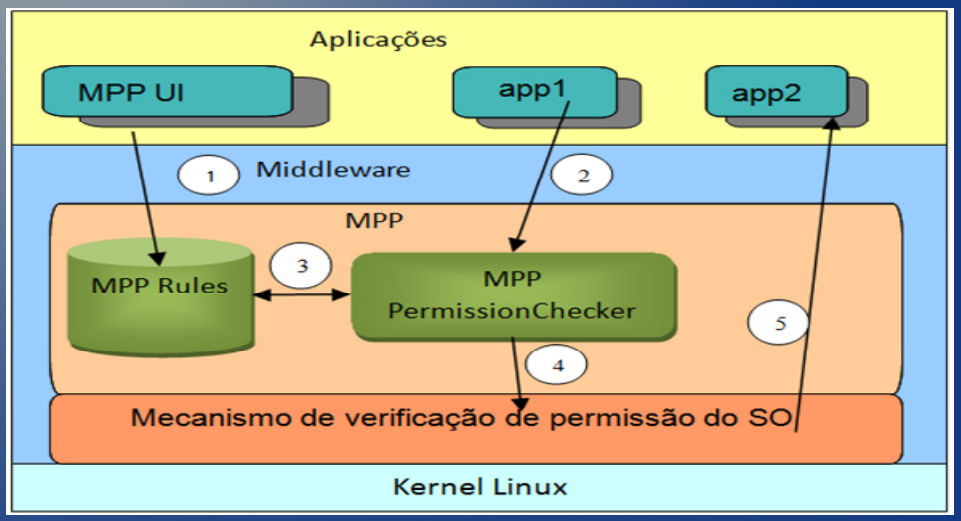
PUPDroid Mechanism

- Allow user control application permission
- Manage how resources and data are accessed
- Use case developed for Android

Implementation

- Implemented with code from google repository
- Developed with Android SDK integrated with Eclipse

Arquitetura



Demo

Conclusão

- Permite definição de acesso personalizado
- Melhor cointrole da privacidade
- Maior segurança