UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROPOSTA DE MÉTODO PARA ANÁLISE PERICIAL EM *SMARTPHONE* COM SISTEMA OPERACIONAL ANDROID

ANDRÉ MORUM DE LIMA SIMÃO

ORIENTADOR: FLÁVIO ELIAS GOMES DE DEUS

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
ÁREA DE CONCENTRAÇÃO INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO

PUBLICAÇÃO: PPGENE.DM – 081/2011
BRASÍLIA / DF: SETEMBRO/2011

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROPOSTA DE MÉTODO PARA ANÁLISE PERICIAL EM *SMARTPHONE* COM SISTEMA OPERACIONAL ANDROID

ANDRÉ MORUM DE LIMA SIMÃO

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE PROFISSIONAL EM INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO.

APROVADA POR:

Flávio Elias Gomes de Deus, Doutor, ENE/FT
(Orientador)

Robson de Oliveira Albuquerque, Doutor, ABIN
(Examinador Interno)

Georges Daniel Amvame Nze, Doutor, FGA-UnB
(Examinador Externo)

DATA: BRASÍLIA/DF, 27 DE SETEMBRO DE 2011.

FICHA CATALOGRÁFICA

SIMÃO, ANDRÉ MORUM DE L.
Proposta de Método para Análise Pericial em *Smartphone* com Sistema Operacional Android [Distrito Federal] 2011.
xiv, 96p, 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2011). Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Perícia Forense
2. Análise de Evidência
3. Telefone Celular
4. *Smartphone*
5. Android
I. ENE/FT/UnB.
II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

SIMÃO, ANDRÉ MORUM DE L. (2011). Proposta de Método para Análise Pericial em *Smartphone* com Sistema Operacional Android. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM – 081/2011, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 96p.

CESSÃO DE DIREITOS

NOME DO AUTOR: André Morum de Lima Simão

TÍTULO DA DISSERTAÇÃO: Proposta de Método para Análise Pericial em *Smartphone* com Sistema Operacional Android.

GRAU: Mestre ANO: 2011

André Morum de Lima Simão
SQS 303, Bloco G, Ap 104, Asa Sul
CEP 70.336-070 – Brasília – DF - Brasil

Dedico este trabalho à minha família, em especial à minha esposa, pessoa com a qual divido todos os meus momentos de felicidade e angústia. E a todos os Peritos Criminais do Brasil que, mesmo sem a devida valorização da carreira, e com todas as adversidades comuns ao serviço público brasileiro, conseguem produzir trabalhos de alto nível e de qualidade incontestável.

AGRADECIMENTOS

Ao meu orientador Prf. Dr. Flávio Elias Gomes de Deus, pela sua capacidade de coordenar e transmitir de forma clara e objetiva os conhecimentos necessários para a realização deste trabalho.

Ao Prof. Msc. Laerte Peotta de Melo, por acreditar na capacidade de seus alunos, pelo incentivo, total disponibilidade e compartilhamento seu conhecimento sobre análise forense, necessários para o desenvolvimento deste trabalho.

Ao colega de trabalho e Prof. Dr. Helvio Ferreira Peixoto que, sem demonstrar qualquer abatimento diante das dificuldades, foi essencial para a realização deste Mestrado, mostrando-se como um exemplo de postura e determinação.

Ao colega de trabalho e amigo Fábio Caús Sícoli, que sempre prestou apoio nos estudos das disciplinas mais complexas, sendo um parceiro exemplar nos trabalhos de classe e um ótimo revisor deste trabalho, tendo sempre a paciência e destreza de um professor.

A todos, os meus sinceros agradecimentos.

O presente trabalho foi realizado com o apoio do Departamento Polícia Federal – DPF, com recursos do Programa Nacional de Segurança Pública com Cidadania – PRONASCI, do Ministério da Justiça.

RESUMO

PROPOSTA DE MÉTODO PARA ANÁLISE PERICIAL EM *SMARTPHONE* COM SISTEMA OPERACIONAL ANDROID

Autor: André Morum de Lima Simão
Orientador: Flávio Elias Gomes de Deus
Programa de Pós-graduação em Engenharia Elétrica
Brasília, setembro de 2011

Existem abordagens periciais bem difundidas e documentadas para exames em aparelhos celulares e computadores, mas não são suficientemente detalhadas para atender as especificidades de um celular com o sistema operacional Android. O objetivo deste trabalho é, a partir das abordagens atuais de análise forense em telefones celulares, propor um método específico para aqueles com o sistema operacional Android, dadas as peculiaridades da plataforma e as situações encontradas pelo analista pericial. Com a crescente adoção do sistema operacional Android nos dispositivos móveis e a própria evolução da plataforma, há uma tendência natural de estes equipamentos conterem cada vez mais informações que podem ser úteis ao processo investigativo.

A partir do método proposto foram mapeadas, por meio da diagramação, as situações reais com que os peritos se deparam durante as etapas de apreensão, aquisição dos dados, exame e documentação, fornecendo os subsídios necessários para realizar os procedimentos forenses da forma correta.

Foram propostos estudos de caso com base em três cenários distintos. Os cenários foram criados a partir de *smartphones* utilizados rotineiramente por usuários com perfis de utilização distintos. Assim, foi possível verificar o trabalho desenvolvido nesta dissertação a partir da aplicação do método em diferentes situações em que o analista pode se deparar.

ABSTRACT

PROPOSED METHOD FOR FORENSIC ANALISYS IN SMARTPHONE WITH ANDROID OPERATING SYSTEM

Author: André Morum de Lima Simão
Supervisor: Flávio Elias Gomes de Deus
Programa de Pós-graduação em Engenharia Elétrica
Brasília, September of 2011

Although there are well documented and widespread approaches about forensic exam on mobile devices and computers, they are not detailed enough to meet all the specificities of an Android phone. The goal this work is, based on the actual guidelines of cell phones forensic analysis, create a specific method for the ones with the Android operating system, given the peculiarities of the platform and the situations that the forensic analyst will face. With the increasing adoption of the Android operating system in mobile devices and the evolution of the platform itself, there is a natural tendency of these devices increasingly contain information that may be useful to the investigation process.

With this method, it was possible to map, through a workflow, real situations that forensic analysts could face in the phases of cell phone seizure, data acquisition, exam and report, giving the necessary knowledge to execute the forensic procedures in a correct way.

Case studies were proposed based on three different scenarios. The scenarios were created from smartphones used routinely by users with different usage profiles. Thus, it was possible to verify the work in this thesis from the application of the method in different situations in which the analyst may come across.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIAÇÕES

A2DP – *Advanced Audio Distribution Profile*

ACPO – *Association of Chief Police Officers*

ANATEL – Agência Nacional de Telecomunicações

APDUs – *Application Protocol Data Units*

API – *Application Program Interface*

AVRCP – *Audio/Video Remote Control Profile*

CDMA – *Code Division Multiple Access*

DHCP – *Dynamic Host Configuration Protocol*

DVM – Dalvik *Virtual Machine*

eMMC – *Embedded MultiMediaCard*

FTK – *Forensic Tool Kit*

FTL – *Flash Translation Layer*

GPS – *Global Positioning System*

GPS – *Global Positioning System*

GSM - *Global System for Mobile Communications* ou *Groupe Spécial Mobile*

HSPA – *High Speed Packet Access*

IBGE – Instituto Brasileiro de Geografia e Estatística

IBM – *International Business Machines*

IMEI – *International Mobile Equipment Identifier*

IrDA – *Infrared Data Association*

ISMI – *Intenational Mobile Subscriber Identity*

JIT – *Just in Time*

JTAG – *Joint Task Action Group*

MTD – *Memory Technology Device*

NFC – *Near Field Communication*

NFI – *Netherlands Forensic Institute*

NIST – *National Institute of Standards and Technology*

OHA – *Open Handset Alliance*

OS – *Operating System*

PIN - *Personal Identification Number*

PUK - PIN *Unlock Key*

RIM – *Research In Motion*

SDK – *Software Development Kit*

SIM - *Subscriber Identity Module*

SIP – *Session Initiation Protocol*

SMS - *Short Message Service*

SO – Sistema Operacional

SSD – *Solid State Disk*

TAC – *Type Allocation Code*

USB – *Universal Serial Bus*

VM – *Virtual Machine*

VoIP – *Voice over Internet Protocol*

VPN – *Virtual Private Network*

YAFFS2 – *Yet Another Flash File System* 2

1. INTRODUÇÃO

O telefone celular é um dispositivo usado comumente pela população brasileira, que se popularizou no decorrer dos anos. Pesquisa realizada pela Anatel (Agência Nacional de Telecomunicações) (Anatel, 2011), informa que no mês de maio de 2011 o Brasil tinha 215 milhões de acessos na telefonia celular, apontando nos primeiros cinco meses do ano um crescimento de 5,95%, indicando que há mais celulares habilitados do que habitantes no país.

Esta mesma pesquisa mostra, um grande aumento do uso da rede mundial de computadores, Internet, por meio da tecnologia de banda larga móvel 3G, com um crescimento neste ano de 27,28%. Com este número, o Brasil totalizava em maio de 2011 185.934.633 (cento e oitenta e cinco milhões, novecentos e trinta e quatro mil e seiscentos e trinta e três) terminais 3G (Anatel, 2011). O sucesso dos dispositivos móveis de acesso à Internet, a exemplo dos *smartphones*, *tablets*, *netbooks* e equipamentos GPS (*Global Positioning System*), mostra exatamente que a convergência de tecnologias vem sendo bem aceita pelo mercado.

A partir daí, com a popularização da Internet, o advento das redes móveis e a disseminação dos dispositivos móveis, nota-se que o mercado, com o apoio do avanço tecnológico, vem apostando cada vez mais em dispositivos que ofereçam convergência destas tecnologias.

Com o avanço tecnológico, o mercado dos telefones celulares evoluiu de tal forma que atualmente têm-se os *smartphones*. Esses são dispositivos que possuem um grande poder computacional, oferecendo aos usuários uma grande diversidade de aplicações que podem utilizar recursos providos pela Internet. Os telefones celulares estão entre os dispositivos mais populares, sendo os *smartphones* entre os objetos de maior desejo daqueles que gostam de tecnologia e buscam facilitar o acesso a diversas fontes de informação.

Neste cenário, os sistemas operacionais voltados para estes dispositivos aumentaram suas funcionalidades a fim de suprir a necessidade do usuário. Vê-se dispositivos com 2GB de memória RAM integrada, aceleradores gráficos e processadores de mais de 1GHz, com núcleo duplo, onde o sistema operacional atua de forma a prover todos os serviços ao usuário.

Conforme citado por Jansen e Ayes (Jansen e Ayers, 2007), a comunidade forense se depara com constantes desafios para se manter atualizada na busca por evidências relevantes à investigação. Os telefones celulares são utilizados por muitas pessoas no seu dia-a-dia, tanto

para fins pessoais como profissionais, sendo uma potencial fonte de informação para um determinado apuratório.

O aumento dos recursos providos pelo SO dos dispositivos móveis, da capacidade de processamento e armazenamento do hardware e a diminuição do seu custo, tornam os *smartphones* grandes provedores de informação. A análise pericial nesse tipo de dispositivo pode trazer muitas informações a respeito do seu usuário, uma vez que funcionalidades como armazenamento de arquivos, histórico de Internet, agenda, contatos, e até mesmo acesso a serviços de computação em nuvem, estarão disponíveis no *smartphone*. Do ponto de vista pericial, à medida que as características do hardware e, principalmente, do software do *smartphone* forem compreendidas, é possível definir como ele deve ser manuseado e onde as informações relevantes estão presentes no dispositivo.

1.1. DEFINIÇÃO DO PROBLEMA

A partir da evolução dos telefones celulares para os chamados *smartphones*, pode-se notar também a mudança nos perfis dos seus usuários. Desta forma são considerados três diferentes grupos (Speckmann, 2008):

- O usuário normal: que utiliza apenas as aplicações básicas do telefone celular, tais como serviço de mensageria SMS (*Short Message Service*), realização de chamadas, calculadora, armazenamento de contatos e despertador;
- O usuário avançado: que utiliza outras aplicações além daquelas fornecidas nativamente[1] pelo telefone celular, integrando-o ao seu cotidiano;
- O usuário expert: que utiliza a maioria dos recursos de hardware e software do aparelho de telefonia celular, com conhecimentos para realizar a instalação de customizações, chegando até mesmo a desenvolver aplicações específicas.

Cada um dos usuários tem necessidades diferentes, sendo que a plataforma Android consegue atingir todos estes grupos de usuários de forma indiscriminada, por conta da utilização de um ambiente altamente amigável, portável e com grande capacidade tecnológica.

[1] No dispositivo móvel com Android pode-se definir aplicações nativas àquelas que vêm instaladas de fábrica no sistema oferecido pela Google, ou seja: Gmail, Gtalk, Android Market, Câmera, Contatos, E-mail Corporativo, Mapas, Relógio, Agenda, YouTube, Pesquisa, Calculadora e Galeria. Estes aplicativos são disponibilizados no emulador do sistema operacional Android disponibilizado no seu SDK (*Software Development Kit*) e podem sofrer alterações a depender da sua versão.

Desta forma, os *smartphones* com a plataforma Android conseguem atingir uma grande parcela dos usuários. Além disso, devido à capacidade da plataforma Android de prover um grande número de funcionalidades, o dispositivo armazena valiosas informações sobre seu proprietário, o que o configura como um repositório de provas para fatos que se queira elucidar ou, simplesmente, mais uma fonte para obter informações a fim de subsidiar uma investigação.

Diferentemente da abordagem de aquisição de dados em ambientes computacionais, em que geralmente os dados são extraídos no estado em que foram encontrados, ficando preservados na sua integralidade, a extração de dados dos *smartphones* normalmente necessita de alguma iteração no dispositivo, que utiliza procedimentos diferentes a depender do sistema operacional utilizado. Além disso, nota-se que utilizam memórias embutidas, cujo acesso, sendo direto ao hardware, é delicado e complexo. Assim, é preciso instalar aplicativos ou utilizar ferramentas diretamente no dispositivo para que se proceda à análise pericial dos dados armazenados de forma mais intuitiva.

Assim, é necessária a criação de um método para realizar a correta extração dos dados destes equipamentos, dada as situações que serão encontradas, devido aos diferentes perfis dos usuários dos *smartphones* Android e das funcionalidades do dispositivo móvel, a fim de evitar que informações importantes à investigação deixem de ser coletadas pela equipe de analistas periciais que realizarão o exame pericial no telefone celular.

Dada a grande variedade de equipamentos que possui o sistema operacional Android, este trabalho limita-se a realizar a pesquisa nos dispositivos mais bem aceitos pelo mercado, ou seja, nos *smartphones*, onde o Android conseguiu se destacar. Isso não significa que o método a ser proposto não possa ser utilizado nos outros dispositivos, entretanto sugere-se sua validação ou aplicação crítica, tendo em vista o foco da dissertação.

1.2. JUSTIFICATIVA

Com relação aos *smartphones*, nota-se um fenômeno que foi análogo ao que ocorreu com os computadores há algumas décadas. Inicialmente, cada telefone celular executava um sistema embarcado próprio e específico. Nos últimos anos, está havendo uma popularização de plataformas comuns de software para diferentes telefones celulares.

Dentre os sistemas operacionais (SO) disponíveis para este tipo de plataforma, estão o iOS (sistema instalado nos dispositivos móveis da Apple), o Windows Phone, Symbian OS para dispositivos Nokia, BlackBerry e o Android OS disponibilizado pela *Open Handset Alliance*. Segundo pesquisas do Gartner Group (Pettey e Stevens, 2011), no final do ano de 2011 o Android OS se tornará o sistema operacional mais popular no mundo, com 38,5% do mercado, com previsões de crescer mais 10,7%, chegando a 49,2% do mercado mundial em 2012. A Figura 1.1 mostra o crescimento da plataforma Android em números absolutos. Outro dado interessante nesta pesquisa é referente à popularização dos *smartphones*, que segundo a Gartner, em 2015 custarão menos de US$300,00, chegando a conquistar 67% do mercado. Nesta pesquisa, foi possível observar que o Android OS é a plataforma que mais cresce. Associando aos dados de outra pesquisa da Gartner feita em 2010 (Pettey e Tudor, 2010), em 2009 o Android OS possuía apenas 3,9% do mercado americano, com os dados na nova pesquisa, tem-se em 2010 o Android OS com 22,7% do mercado já conquistado a nível mundial, um crescimento considerável dado um período de apenas um ano.

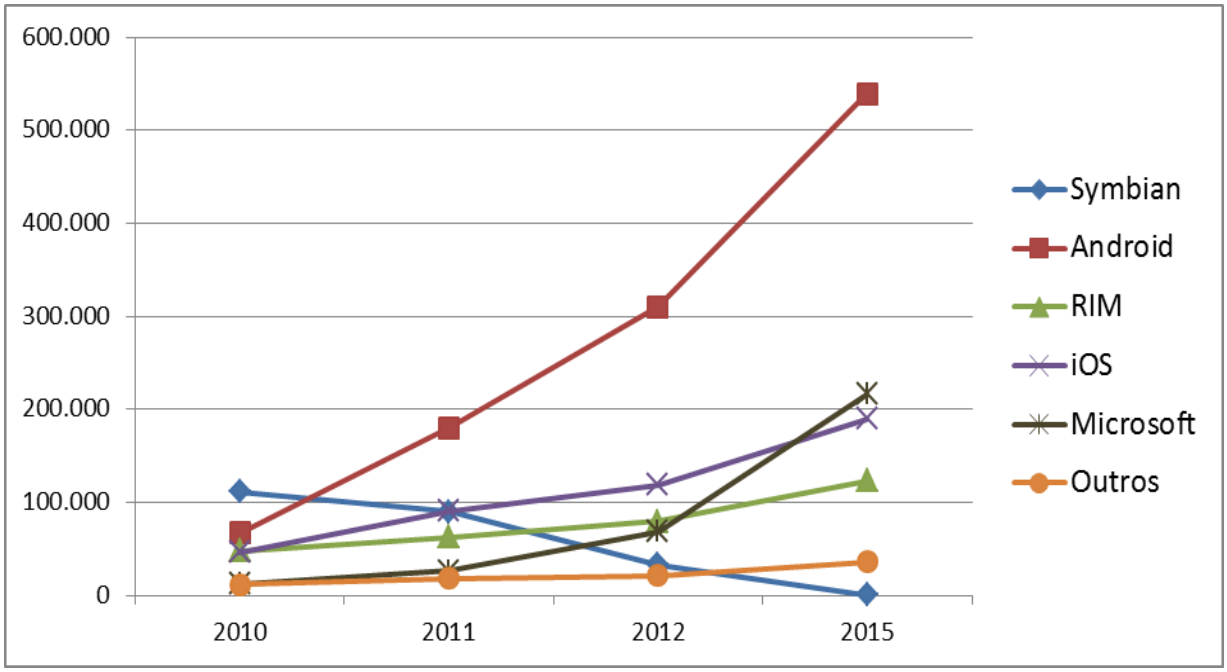


Figura 1.1 - Evolução e previsões de vendas de *smartphone*s com os sistemas operacionais descritos em números absolutos[2].

Conjectura-se o crescimento da plataforma Android à sua capacidade de customização e suporte aos mais modernos recursos e aplicativos disponíveis para estes tipos de dispositivos.

[2] Dados obtidos da pesquisa da Gartner (Pettey e Stevens, 2011)

Trata-a de um sistema aberto baseado em Linux, kernel 2.6, desde sua versão 1.5, disponibilizada por grandes empresas de TI do mercado, que se uniram em um consórcio chamado *Open Handset Alliance* (OHA), liderado pela Google. Além disso, possui diversos recursos, a exemplo de suporte a multimídia, Java, aplicações 3D, SQLite, geolocalização, GSM (*Global System for Mobile Communications*), navegação web (Google Inc, 2011f). Tem-se também nesta plataforma uma interface muito amigável e de fácil portabilidade entre diversos hardwares dos diferentes fabricantes de dispositivos móveis.

A partir do que foi demonstrado, temos na plataforma Android um *smartphone* com grande aceitação no mercado mundial. Do ponto de vista pericial, vemos que a análise forense de telefones celulares é a ciência de recuperação de informações digitais destes dispositivos que, assim como a ciência forense como um todo, tem que utilizar técnicas bem fundamentadas e aceitas pela comunidade forense. Assim, a criação de um método de análise forense de dispositivos móveis com sistema operacional Android, baseado em modelos já aceitos pela comunidade, descrevendo as peculiaridades do sistema, visa auxiliar o Perito a atuar de forma correta com este tipo de dispositivo móvel, orientando-o a buscar as informações disponíveis no sistema da forma menos invasiva possível, por meio de um método forense voltado para ambiente móvel, a fim de que a evidência possa ser extraída e analisada de forma a ser útil e incontestável para o processo investigativo.

Segundo o NIST (*National Institute of Standards and Technology*), a análise forense em telefones celulares deve passar pelos processos de apreensão, aquisição, exame e relatório (Jansen e Ayers, 2007). Assim, algumas técnicas devem ser estudadas e usadas, a depender da configuração do dispositivo, a fim de se criar um método para analisar as informações presentes.

Serão estudadas as características da plataforma Android sob o ponto de vista pericial, nas etapas de apreensão, aquisição, exames e documentação (geração do relatório/laudo). Também serão avaliados os aplicativos instalados na plataforma. Será definido um método para extração e análise das informações úteis do sistema operacional. O método será composto por um conjunto de regras e procedimentos que tem como finalidade obter a informação armazenada no dispositivo, dada as diferentes situações que o analista pode se deparar, a exemplo de *smartphone* desbloqueado de um usuário comum, ou um *smartphone* com permissões de super usuário, ou com acesso de depuração via USB (*Universal Serial Bus*) ativado.

A partir da situação encontrada, será realizada a extração das informações que puderem ser acessadas, a exemplo dos históricos de Internet, mensagens de texto, mensagens eletrônicas, lista de contatos e sua ligação com possíveis redes sociais, registro de chamadas, imagens, dentre outras, por meio do uso de técnicas que possam ser adotadas como padrões para análise pericial desses dispositivos.

1.3. OBJETIVO DA DISSERTAÇÃO

O objetivo desta dissertação é criar um método para análise pericial de dispositivos *smartphone* com o sistema Android, a partir da identificação das diferentes situações que o analista pericial pode se deparar nos processos de apreensão, aquisição dos dados, exame e documentação. A extração das informações deverá ser realizada de tal forma a obter aquilo que é possível a partir da configuração e dos dados armazenados no equipamento.

Especificamente, esta dissertação objetiva:

- Analisar as abordagens documentadas atualmente sobre forense em telefones celulares;
- Identificar e descrever as características do Sistema Operacional Android sob uma perspectiva forense;
- Descrever técnicas para análise pericial de *smartphones* Android;
- Mapear as diferentes situações que um analista pericial pode encontrar ao examinar um *smartphone* com sistema operacional Android;
- Definir quais procedimentos a serem adotados para cada etapa do processo de análise do *smartphone* Android.

Serão mostradas que as abordagens existentes até a conclusão deste tabalho para forense em telefones celulares são deficientes ou incompletas a fim de se realizar uma análise pericial em um *smartphone* com o sistema operacional Android. Assim, será proposto um método para orientar o analista pericial, que terá como missão extrair as informações do dispositivo móvel Android, fornecendo para equipe de investigação a informação possível de ser obtida, dada à situação em que o mesmo foi encaminhado para análise.

1.4. METODOLOGIA

Para proposição do método criado neste trabalho, optou-se por incialmente realizar uma revisão da literatura sobre análise pericial em telefones celulares e dispositivos móveis.

Algumas das abordagens estudadas partem de premissas definidas para exames em computadores, o que prejudica a análise em um *smartphone* Android, pois a interação com o dispositivo geralmente é necessária. Isso impossibilita uma preservação da evidência na sua integralidade. Outro fator observado é que os estudos relacionados à forense em telefones celulares é que eles normalmente partem de aspectos gerais, devido a grande variedade de fabricantes e modelos, perdendo aspectos específicos de cada plataforma.

A partir de então, foram estudadas as características da plataforma Android voltada para *smartphones*. Esse estudo visou apresentar das características gerais do sistema operacional, os aplicativos e algumas ferramentas, sob uma ótica forense. Pode-se, desta forma, verificar suas singularidades a fim de identificar os fatores que pudessem ser abordados na proposição do método específico para análise pericial desta plataforma.

Logo, o método proposto para realizar uma análise pericial em um *smartphone* com sistema operacional Android é baseado nas melhores práticas utilizadas atualmente pela Polícia Federal do Brasil, pelo NIST (Jansen e Ayers, 2007), pelo Departamento de Justiça dos Estados Unidos (Ashcroft, 2001), polícia Inglesa (Association of Chief Police Officers, 2008) e Instituto Forense da Holanda (Netherlands Forensic Institute, 2007), e busca diferenciar-se adaptando essas melhores práticas às peculiaridades da plataforma Android.

1.5. ORGANIZAÇÃO DO TRABALHO

Ante o exposto, a dissertação foi dividida em 6 (seis) capítulos. Neste capítulo foi realizada uma introdução do trabalho, onde é definido o problema a ser resolvido, sua justificativa e os objetivos do trabalho. O capítulo 2 fornece uma visão das abordagens usadas atualmente nos Estados Unidos, Inglaterra e Holanda, para análise de telefones celulares, atentando para o fato da apreensão, aquisição da informação, exame e documentação do processo forense.

No capítulo 3 é descrita a plataforma Android, suas principais características, suas versões e suas funcionalidades, fornecendo uma base para entendimento do sistema operacional. Apresenta também a estrutura do sistema e da plataforma sob o ponto de visa forense, com detalhes sobre o funcionamento do sistema e consequentemente, onde os dados estão armazenados e como são acessados.

O capítulo 4 apresenta de forma clara, objetiva e diagramada o método proposto para extração da informação do *smartphone* com o sistema operacional Android, com a finalidade de

detalhar ao analista pericial como deve proceder nas etapas de apreensão, aquisição dos dados, exames e geração do relatório/laudo (documentação);

No capítulo 5 serão apresentados estudos de caso, aplicando o método proposto em três cenários distintos. Um com o celular ligado, sem bloqueio e sem permissões de super usuário; outro cenário com celular desligado, com bloqueio e sem acesso de depuração e; no último cenário um celular bloqueado, com acesso de depuração restrito, e com permissões super usuário.

Finalmente, o capítulo 6: conclui o trabalho, apresentando uma síntese dos resultados obtidos a partir da validação do método proposto, descrevendo sua relevância. Também são propostos trabalhos futuros, as dificuldades encontradas, assim como apresenta formas de se dar continuidade ao trabalho desenvolvido.

2. AS ABORDAGENS DE FORENSE EM TELEFONES CELULARES

Forense significa a arte ou estudo do discurso argumentativo em que a ciência é usada para fornecer ou demonstrar fatos, como por exemplo, aplicar a ciência ao direito (Owen, Thomas e Mcphee, 2010). Analistas forenses extraem, analisam, identificam e reconstroem evidências a fim de obter conclusões a respeito de um fato, usando a ciência para fornecer o devido embasamento na conclusão das suas argumentações.

As melhores práticas de análise forense em telefones celulares definem procedimentos para apreensão, aquisição, exame e documentação (geração do relatório/laudo). Estas etapas são importantes, entretanto, existe um histórico de terem sido definidas a partir de procedimentos utilizados em forense de computadores (Owen, Thomas e Mcphee, 2010), a exemplo da abordagem da *Association of Chief Police Officers* (ACPO), partindo de regras gerais para aquisição da informação em equipamentos computacionais com características muito similares.

Os telefones celulares estão em momento tecnológico de grande evolução, possuindo características diferentes a depender do fabricante, do modelo e do sistema operacional como: diferentes memórias para armazenamento do sistema e dos dados; diferentes tipos de hardware; diferentes formas de conexão (microUSB, miniUSB, bluetooth, infra vermelho e padrões proprietários) e; diferentes sistemas operacionais, a exemplo de aparelhos com customizações do sistema operacional Linux. Assim, ao contrário das abordagens periciais em computadores, que já estão bem documentadas e difundidas, é interessante descrever os procedimentos forenses focando as peculiaridades dos telefones celulares, observando também a grande variedade de equipamentos disponíveis atualmente.

Como descreve a ACPO em seu segundo princípio, em um exame mais específico onde há necessidade de extração da informação direta do dispositivo, o examinador deve ter as competências e expertise necessárias a fim de obter a informação e explicar a relevância e implicações dos procedimentos utilizados. Como os telefones celulares possuem diferentes softwares, hardwares e funcionalidades, vê-se a necessidade de escrever procedimentos específicos para diferentes categorias de aparelhos.

Neste capítulo são estudadas as abordagens utilizadas atualmente para forense em telefones celulares, em especial a do NIST (Jansen e Ayers, 2007), Departamento de Justiça dos

Estados Unidos (Ashcroft, 2001), ACPO (Association of Chief Police Officers, 2008) e NFI (Netherlands Forensic Institute, 2007), uma vez que possuem procedimentos que são os mais utilizados e aceitos atualmente, inclusive pela Polícia Federal do Brasil. Conhecendo estas abordagens, é possível propor um método para análise forense específico para a plataforma Android, abrangendo as peculiaridades desse ambiente.

2.1. PROCEDIMENTOS DE BUSCA, APREENSÃO E PRESERVAÇÃO DO TELEFONE CELULAR

Antes de efetivamente extrair os dados dos telefones celulares, deve-se fazer a correta preservação do dispositivo para que chegue a um analista pericial na melhor condição possível para se realizar o exame. A apreensão tem por objetivo preservar as evidências de tal forma a evitar a perda ou alteração da prova a ser apreendida. Também envolve a busca por mídias eletrônicas que possam possuir informação útil a respeito do que esta sendo investigado. A questão mais importante é preservar de forma adequada os dispositivos que forem apreendidos, documentando-os conforme preconiza o Código de Processo Penal Brasileiro (Brasil, 2003) e os normativos vigentes (DITEC/DPF, 2010).

Segundo o Departamento de Justiça Norte Americano (Ashcroft, 2001), na etapa de apreensão, a equipe tem o dever de avaliar e preservar a cena, documentá-la, coletar as evidências, realizar o acondicionamento, transporte e armazenamento da evidência de forma confiável, evitando danificá-la, primando pela sua preservação.

Tanto o NIST (Jansen e Ayers, 2007), a ACPO (Association of Chief Police Officers, 2008), quanto o NFI (Netherlands Forensic Institute, 2007), descrevem que ao se deparar com dispositivos móveis, deve-se lidar com a evidência de acordo com o que está sendo apurado. Por exemplo, deve-se atentar para o fato que o dispositivo pode conter evidências como DNA ou impressões digitais, que podem ser destruídas ou contaminadas se este for indevidamente manuseado. Deve-se também observar o fato que há casos em que a tela de descanso é ativada ou o telefone é desligado e, para se obter acesso ao telefone, pode ser necessário passar por um sistema de autenticação, devendo assim avaliar a questão do desligamento ou não do equipamento, devendo acondicioná-lo de tal forma que não sofra acionamento das teclas de forma acidental. Deve-se também observar os acessórios do equipamento, uma vez que não há padronização, por exemplo, do carregador.

Apesar de o NIST recomendar evitar entrar no local com dispositivos Wi-Fi e Bluetooth ativados, para não ocorrer interações indesejadas com o objeto da busca (Jansen e Ayers, 2007), o analista pericial pode avaliar a utilização destes recursos para buscar por equipamentos que possam estar utilizando destas tecnologias para facilitar a documentação e até mesmo encontrar um equipamento escondido. Também devem ser arrecadados os acessórios dos dispositivos, como baterias, carregadores, e *docking stations*, principalmente quando se tratar de um telefone celular mais simples ou com baixa aceitação no mercado. É interessante buscar por cartões SIM (*Subscriber Identity Module*) e memórias removíveis, pois estas podem conter informações úteis ao apuratório. Deve-se entrevistar os donos dos equipamentos arrecadados a fim de obter possíveis senhas de acesso.

No processo de busca e apreensão, a cena poderá ser fotografada e, se houver a presença de um perito na equipe, em havendo necessidade, será realizado um laudo de local onde constarão informações acerca de todas as evidências coletadas. Deve-se atentar também para os dispositivos móveis que estiverem conectados às *docking stations* ou ao computador, pois pode estar ocorrendo transferências de dados que, caso sejam interrompidas, poderão cessar a transferência de dados ou sincronização (Jansen e Ayers, 2007).

As abordagens empregadas pelo NIST, ACPO e NFI descrevem a importância de isolar o dispositivo móvel da rede de comunicação. Assim, evita-se que dados recebidos pelo dispositivo após a sua apreensão não sobrescrevam dados já existentes. Pode-se citar o exemplo das mensagens de texto SMS que, em alguns modelos de telefones celulares, sobrescrevem automaticamente as mais antigas quando uma nova mensagem chega. Assim, pode-se utilizar um invólucro que bloqueia o recebimento dos dados para acondicionamento ou deve-se desligar o aparelho no momento em que for apreendido. Outra opção seria ativar o modo avião (*offline*) nos dispositivos que tiverem tal opção, a fim de evitar o desligamento completo do equipamento, inclusive economizando o consumo de bateria.

Cada uma das três alternativas tem suas vantagens e desvantagens, que devem ser levadas em consideração a depender do caso ou do alvo investigado. O desligamento do dispositivo pode dificultar o seu acesso quando da realização dos exames, uma vez que códigos de autenticação podem ser solicitados quando do seu religamento. Já o isolamento em uma sacola de bloqueio de sinal, pode aumentar significativamente o consumo da bateria do dispositivo, uma vez que o mesmo aumentará a potência de sua antena para tentar encontrar uma torre mais distante (Association of Chief Police Officers, 2008). Finalizando, a ativação do modo avião exigirá

uma interação de uma agente da lei com o dispositivo, sendo que nem sempre esta pessoa está habilitada para realizá-la, o que feriria uma das premissas da preservação, uma vez que o dispositivo só poderia ser manuseado por pessoa habilitada, ou até mesmo uma interação antes da realização dos exames pode não ser adequada.

2.2. AQUISIÇÃO DOS DADOS

A aquisição consiste em extrair do telefone celular a informação para posterior análise. A aquisição é idealmente realizada em um ambiente isolado da rede de comunicação do dispositivo, como um laboratório, por meio do uso de hardware e softwares adequados para obtenção dos dados.

Antes de realizar a extração dos dados, o examinador deve evitar que o dispositivo se comunique com a rede de telefonia ou realize conexões com a rede Wi-Fi, Bluetooth, IrDA (infra vermelho). Para tanto, pode-se utilizar equipamentos específicos para isolar tal comunicação ou realizar uma intervenção direta no dispositivo, desabilitando tais serviços, atentando para o que já foi discutido na fase de apreensão e preservação.

O perito examinador deve iniciar os trabalhos de extração preferencialmente com a bateria do telefone celular totalmente carregada e, quando for o caso, com uma fonte direta de energia conectada. Desta forma evita-se a corrupção ou perda dos dados durante o processo.

Para realizar a extração de forma adequada, deve-se observar que os dispositivos atuais possuem, além de memória interna, cartões de memória, e, a depender do dispositivo, dados armazenados na Internet por meio de aplicativos instalados (computação em nuvem), que não são citadas nas abordagens usadas atualmente. Assim o perito examinador deve levar em conta o objetivo dos exames que serão realizados, para avaliar até onde poderá extrair informações que se encontram disponíveis na nuvem (Internet) e são disponibilizadas por meio do dispositivo.

Normalmente, os dados armazenados localmente no dispositivo, principalmente aqueles disponíveis em sua memória interna, são extraídos com o uso de softwares forenses específicos para tal finalidade. A interação do telefone celular com o software forense deve ser a menor possível, por isso deve-se tentar estabelecer a conexão primeiramente via cabo (USB ou portas seriais ou paralelas), depois infravermelho, bluetooth e por último Wi-Fi (Association of Chief Police Officers, 2008). Em algumas situações, tais softwares podem

não funcionar adequadamente em alguns equipamentos, sendo necessária a extração com a utilização dos aplicativos proprietários dos fabricantes do dispositivo móvel ou uma extração manual do conteúdo, que deve ser realizada por analista pericial com conhecimentos específicos sobre a plataforma do telefone celular em questão.

Também é importante a correta identificação do equipamento a ser periciado. A descrição do equipamento com dados como fabricante, marca, modelo e operadora podem ajudar na extração dos dados do dispositivo e na sua cadeia de custódia[3]. A identificação das interfaces usadas pelo equipamento ajuda a utilização dos cabos e softwares corretos para obtenção dos dados. A identificação dos telefones celulares pelo IMEI (*International Mobile Equipment Identifier*) faz-se necessária uma vez que esta numeração consiste em um número de 15 dígitos, cujos 8 iniciais indicam o TAC (*Type Allocation Code*), fornecendo o modelo e a origem, e os demais dígitos são de uso fabricante (Jansen e Ayers, 2007).

Outro identificador importante é o número do cartão SIM (*Subscriber Identity Module*). O chip ou cartão SIM é um cartão inteligente que possui microprocessador, usado para implementar segurança (autenticação e geração de chaves criptográficas) (Quirke, 2004). Além das informações de habilitação da rede de telefonia celular contidas em um chip, este é capaz de armazenar dados correspondentes à agenda telefônica, últimas chamadas, mensagens de texto, dentre outros. Cada chip possui um código IMSI (*International Mobile Subscriber Identity*), que é um código único, de 15 dígitos, utilizado para identificar um único usuário em uma rede GSM (*Global System for Mobile Communications* / *Groupe Spécial Mobile*) (Quirke, 2004).

2.2.1. Considerações sobre as ferramentas de extração

Segundo o NIST (Jansen e Ayers, 2007), alguns critérios são fundamentais para ferramentas forenses, devendo: apresentar os dados de tal forma que sejam úteis e necessários ao investigador, com a finalidade de determinar ou não autoria e culpabilidade; ser precisa, determinística, apresentando os mesmos resultados dada a mesma entrada, e; verificável, garantindo a precisão da saída, fornecendo acesso as etapas intermediárias e apresentação dos resultados.

[3] A Cadeia de Custódia é um processo usado para manter e documentar a história cronológica da evidência, para garantir a idoneidade e o rastreamento das evidências utilizadas em processos judiciais (Lopes, Gabriel e Bareta, 2007).

Ademais, acrescenta-se a estes critérios a compatibilidade, que é a capacidade de agregar o maior número possível de equipamentos móveis disponíveis no mercado, e a atualização, que é a habilidade de se manter atualizada, haja vista que novos dispositivos são disponibilizados no mercado.

Devido à grande diversidade de dispositivos, modelos, versões e fabricantes, e à necessidade do mercado de ter ferramentas forenses atualizadas e compatíveis com sua realidade, as ferramentas forenses devem ser validadas por uma equipe de examinadores. Pode haver situações em que uma determinada ferramenta pode ser muito útil na extração dos dados de uma agenda, entretanto pode falhar na recuperação das dadas dos registros e até mesmo conseguir extrair com sucesso os dados de um modelo específico e não obter sucesso em outros modelos. Ademais, com a chegada ao mercado de celulares sem fabricantes conhecidos, e de baixo custo, a adequabilidade e compatibilidade das ferramentas forenses podem não conseguir acompanhar a realidade mercado, devendo o examinador conhecer a ferramenta forense, estando apto a observar comportamentos não desejáveis que não se adequem aos critérios definidos.

2.2.2. Considerações sobre as memórias dos dispositivos móveis

Os equipamentos móveis possuem em sua estrutura de hardware memórias voláteis e não-voláteis. Toda estrutura do sistema operacional do dispositivo utiliza memória para armazenar dados relativos aos aplicativos instalados, assim como informações relativas ao próprio SO.

Cada fabricante e modelo podem utilizar uma versão de sistema operacional, alterando a forma com que são armazenadas informações de agenda, textuais, imagens, vídeos, calendários e registros de chamadas. Informações estas que usualmente são os focos das extrações. Resumindo, os dados armazenados na memória dos telefones celulares nem sempre estão armazenados da mesma forma, mudando de acordo com o fabricante e modelo do telefone celular.

Os dispositivos com o sistema operacional Android instalado apresentam mesma configuração no gerenciamento da memória. O próprio sistema só pode ser utilizado no dispositivo se o fabricante do hardware atender pré-requisitos para seu correto funcionamento (Google Inc, 2011d). Desta forma, a estrutura com que os dados são armazenados nos equipamentos passa a ser uniformizada, podendo sofrer apenas algumas pequenas alterações de funcionalidades a depender da versão do sistema Android.

2.2.3. Considerações sobre aquisição dos dados

No processo de aquisição lógica dos dados de um telefone celular, este tem a necessidade de estar ligado. Quando se liga o equipamento, estruturas de hardware e software são requeridas para que o processo de inicialização possa ser realizado. Assim, estruturas como registradores e memória são utilizadas, o que provoca alteração nos dados armazenados no dispositivo. Desta forma, quando se realiza a aquisição lógica de um dispositivo móvel que se encontrava desligado, necessariamente o processo provocará alteração de algumas informações disponíveis na memória, uma vez que o telefone celular terá que ser ligado. O examinador deve ter ciência de quais dados podem ou não ser alterados, de tal forma que tente evitar danificar o que pode ser interessante ao apuratório. A extração deve ser realizada com o intuito de se modificar o mínimo possível a informação armazenada no equipamento e, quando os dados tiverem que ser modificados, o examinador deve ter o conhecimento suficiente para saber se o dado a ser alterado é importante ou não para o caso, fazendo as devidas intervenções e documentando o processo.

As ferramentas de extração ajudam o especialista durante o processo de aquisição. Contudo, cabe ao analista pericial conferir se os dados foram devidamente extraídos (Jansen e Ayers, 2007). Uma informação relevante que deve ser levada em conta e conferida são as datas configuradas nos equipamentos móveis. Alguns dispositivos recebem a data diretamente da operadora de telefonia e outros permitem ao usuário inserir tais informações. Atenta-se ainda ao fato de que alguns equipamentos, quando tem sua fonte de energia esgotada ou desconectada, simplesmente retornam a data e hora para o padrão de fábrica. Outros dados além da data e hora também devem ser verificados pelo examinador, uma vez que as ferramentas nem sempre extraem os dados na sua totalidade. Nesta situação, uma aquisição manual através da navegação direta no dispositivo é necessária, devendo o examinador ter cautela para não modificar ou excluir dados importantes da memória do equipamento, mantendo ao máximo a integridade do aparelho (Association of Chief Police Officers, 2008).

Nos casos de dispositivos que utilizam a rede GSM, o cartão SIM também é objeto de aquisição de dados, pois armazena dados de agenda e mensagens SMS (*Short Message Service*). A melhor forma de se obter estes dados é através de um leitor de cartões SIM. Uma ferramenta forense de aquisição pode enviar diretivas APDUs (*Application Protocol Data Units*) diretamente para o cartão, extraindo os dados logicamente, atentando para o status dos códigos de segurança PIN (*Personal Identification Number*) e PUK (PIN *Unlock Key*). A

extração física dos dados é um procedimento mais complexo, uma vez que o cartão possui recursos de proteção.

Tanto a obtenção e análise dos dados do cartão SIM como a extração física dos dados de um telefone celular não são objetos do método proposto neste trabalho. Os métodos de extração do cartão SIM se encontram bem documentados e seguem padrões bem definidos. Já a extração física dos dados de um telefone celular é algo que pode ser muito complexo, pois depende de características de hardware do dispositivo, variando significativamente o método empregado a depender do fabricante e modelo.

2.2.4. Códigos de segurança e controle de aceso

Muitos dispositivos móveis possuem formas de controle de acesso às suas funcionalidades. Os dispositivos GSM com PIN ativado é um tipo de dispositivo bloqueado, cujo acesso só é concedido àquelas pessoas que conhecem a senha de quatro dígitos a ser fornecida. Além de bloqueios fornecidos pelo cartão SIM, há também aqueles fornecidos pelo sistema operacional do equipamento. Entre exemplos de bloqueios realizados pelo sistema operacional, está o código de acesso do iOS da Apple no iPhone e o uso da combinação tátil do SO Android e até mesmo o uso de biometria com o reconhecimento da impressão datiloscópica.

Um cartão SIM bloqueado só poderá ser acessado através da digitação do código PIN, que normalmente é configurado pelo usuário, e, caso este código seja digitado incorretamente por três vezes, o cartão solicitará o código PUK. O PUK é pré-configurado de fábrica e é fornecido no momento da aquisição do cartão SIM. Outra forma de se obter o PUK é por meio de informação da operadora de telefonia ao qual o cartão está vinculado. Caso o PUK seja digitado errado dez vezes o cartão SIM é definitivamente bloqueado. Assim, é recomendável que o analista pericial verifique a quantidade de vezes que o PIN foi digitado incorretamente, e apenas tente a utilização de códigos PIN padrão se não restar apenas uma tentativa de erro. Com relação ao código PUK, o recomendado é não realizar nenhuma tentativa, pois pode levar o cartão SIM a um bloqueio definitivo (Association of Chief Police Officers, 2008).

É importante o examinador se atentar ao fato de manusear o mínimo possível o equipamento a ter os dados extraídos. Alguns dispositivos, ao terem seu cartão SIM substituído, perdem algumas informações que estavam armazenadas em sua memória.

Já os controles de acesso fornecidos em nível de sistema operacional, em alguns modelos de dispositivos, podem ser contornados a fim de fornecer o acesso ao telefone. Desta forma, caso o examinador se depare com este tipo de bloqueio deve pesquisar sobre o modelo do dispositivo e seu SO, a fim de tentar burlar o sistema de controle de acesso, ou buscar soluções alternativas, a exemplo de interrogar o proprietário do equipamento.

Normalmente estes controles vêm desativados de fábrica e, normalmente, o usuário não os habilita, pois gera um nível de desconforto ao utilizar o equipamento. Entretanto, as formas de controle de acesso vêm evoluindo, assim como a importância dos dados armazenados nos dispositivos móveis. Os usuários tendem a buscar mecanismos eficientes e ajustáveis às suas necessidades. Atualmente vemos equipamentos bloqueados com código de segurança PIN, até a utilização de impressões papilares e padrões visíveis (combinação táctil), fornecidos em nível de sistema operacional.

Existem três formas de obter os dados de dispositivos bloqueados: investigando; através de software ou; através de hardware (Jansen e Ayers, 2007). Tentar burlar o mecanismo de acesso diretamente no dispositivo apreendido nem sempre é recomendado, sendo a situação ideal, realizar os procedimentos em um equipamento da mesma marca e modelo daquele que foi arrecadado. Assim pode-se evitar que ações que provoquem perda de dados ou bloqueio do dispositivo ocorram no equipamento original. Exemplo disso é o bloqueio do código PIN ao ser digitado incorretamente três vezes e o bloqueio do acesso ao cartão SIM ao digitar o PUK incorretamente 10 vezes.

Segundo Jansen e Ayers, os métodos baseados em hardware e software são específicos para cada dispositivo onde algumas abordagens devem ser observadas a exemplo de contatar o fabricante ou a operadora de telefonia, a fim de descobrir possíveis *backdoors* ou vulnerabilidades do equipamento em questão; verificar a documentação do dispositivo fornecida pelos fabricantes para saber quais medidas tomar; verificar se existem profissionais no mercado especializados em recuperação das evidências; contatar equipes de assistência técnica especializada a fins de descobrir se há alguma forma de obter as evidências a partir de informações técnicas do equipamento.

A técnica investigativa pode ser bem utilizada pela equipe que vai a campo arrecadar o material a ser examinado. Quando o equipamento é apreendido e seu suposto dono encontra-se no local, é prudente perguntar ao suspeito possíveis senhas que sejam utilizadas para ter

acesso ao dispositivo. Na busca e apreensão é o momento em que o suspeito normalmente não se deu conta do que está ocorrendo e facilita o fornecimento de informações desta natureza.

Outra forma de obtenção de senhas é buscar por informações escritas em pedaços de papel ou coladas no equipamento, dentro da carteira do suspeito, etc. Em último caso, podem-se tentar senhas comuns ao proprietário do telefone celular, como datas importantes, e senhas padrão no caso de PIN. Pode-se também buscar junto às operadoras de telefonia, códigos de desbloqueio, a exemplo do PUK, citado anteriormente.

Outras abordagens usam procedimentos específicos, em nível de software, para burlar o sistema de autenticação dos dispositivos. Esta técnica pode variar muito a depender da marca, modelo e versão do dispositivo e do sistema operacional instalado. No caso dos celulares com o sistema operacional Android, por exemplo, há um aplicativo que consegue burlar o sistema de autenticação padrão, desde que seja possível instalá-lo (Cannon, 2011).

Uma forma mais agressiva de se burlar os mecanismos de controle de acesso é a aquisição por hardware, que apesar de não ser objeto deste trabalho é importante ser citada. Esta técnica também varia muito para cada equipamento e é normalmente usada em conjunto com a técnica de software para obtenção dos dados. Para se realizar o acesso por hardware, o especialista busca possíveis vulnerabilidades em interfaces de configuração, teste e manutenção, para ter acesso à memória principal. Um exemplo é a interface JTAG (*Joint Task Action Group*), que alguns dispositivos utilizam.

Outra forma seria ler o chip de memória diretamente do circuito de memória (Knijff, 2001). Nesta abordagem, a especificação de hardware do equipamento influencia na forma como os dados serão obtidos, sendo mais complexa tanto no processo de obtenção como no de análise dos dados obtidos, uma vez que deverão ser reorganizados em um ambiente diferente do disponibilizado pelo dispositivo móvel.

2.2.5. Considerações sobre memórias removíveis

As memórias removíveis são muito utilizadas nos telefones celulares atuais. Isso se deve em razão do custo deste tipo de memória ter reduzido, assim como sua capacidade de armazenamento ter aumentado e seu tamanho diminuído. Com tamanhos reduzidos, que não ocupam muito espaço físico no telefone celular, e grande capacidade de armazenamento

(podendo superar 32GiB), as memórias removíveis são a opção para se armazenar grande parte dos dados do dispositivo móvel.

A aquisição das informações presentes nestas memórias pode se dar através de ferramentas utilizadas comumente em perícias de computadores. Nem todas as ferramentas forenses de extração de dados de telefone celular possuem a capacidade de obter as informações armazenadas no cartão de memória (Jansen e Ayers, 2007), devendo o examinador saber o funcionamento destas ferramentas e complementar a aquisição dos dados, quando for o caso, utilizando outros meios.

Deve-se atentar que as boas práticas exigem que na aquisição dos dados de memórias removíveis, utilize-se leitor de cartões compatíveis com a mídia em questão, devendo o examinador proteger a memória da escrita indesejada usando bloqueadores de escrita, seja por software ou por hardware.

2.3. O EXAME DE UM TELEFONE CELULAR

O exame é a etapa onde o analista pericial extraíra as informações relevantes dos dados que foram adquiridos na etapa anterior. Há a necessidade que o especialista tenha o conhecimento necessário para lidar com a evidência, devendo ter treinamento específico para esta finalidade (Ashcroft, 2001). Os exames realizados em telefones celulares devem ter um objetivo claro daquilo que se está buscando, onde o examinador deve ter conhecimento do caso em questão. Caso contrário, o relatório (laudo) descrevendo o resultado dos exames não passará de uma simples extração das informações que estavam no telefone celular para outro tipo de suporte, a exemplo do papel ou uma mídia ótica.

Observando o que ocorre atualmente no processo investigativo brasileiro, a autoridade que realiza a solicitação formal do exame deve buscar esclarecer ao máximo em tal documento o motivo que ensejou o pedido. Há casos que, a depender daquilo que se está sendo apurado, o exame pode seguir paradigmas diferentes na análise dos dados extraídos do telefone celular. Por exemplo, em um caso de abuso sexual infantil, o examinador deve iniciar o exame buscando imagens e vídeos que possam ter relação com o objeto da solicitação. Já em casos de tráfico de drogas, a troca de mensagens e relação dos contatos pode ser o primeiro passo.

É importante esclarecer que exame em telefones celulares visa buscar evidências que podem ou não estar relacionada com o caso, e também complementá-lo, ou ajudar em outros exames

que podem vir a ser solicitados. Exemplos desta situação são as anotações de senhas e e-mails que podem ser usados *a posteriori*.

A individualização do telefone celular, associando-o ao seu proprietário é importante e desejável. A partir da identificação do proprietário, seja no momento da aquisição e identificação ou no momento do exame, buscando provas que apontem a propriedade para um determinado indivíduo, é possível determinar autoria. Deve-se considerar que há situações em que o proprietário da linha pode não ser o usuário do telefone celular, podendo esse ser um parente, sua linha clonada ou o cartão SIM furtado (no caso de tecnologia GSM). É importante a caracterização de propriedade por meios mais seguros e eficazes, a exemplo da individualização do telefone celular pela análise dos dados extraídos. No caso de telefones celulares com a plataforma Android, possivelmente haverá uma conta dos serviços da Google associado ao dispositivo, podendo o investigador obter informações individualizadoras, a exemplo do e-mail pessoal e fotos armazenadas no dispositivo.

Na análise dos dados extraídos, o examinador deve basicamente se atentar para configurações como data e hora, linguagem, configurações regionais, contatos, agenda, mensagens textuais, chamadas (realizadas, recebidas e não atendidas), imagens, vídeos, áudio e mensagens multimídias (Jansen e Ayers, 2007). Entretanto, com a evolução dos telefones celulares, a complexidade dos exames tem aumentado, uma vez que os analistas devem buscar também e-mails, históricos de navegação web, documentos, informações do GPS, aplicativos específicos, informações relativas à computação em nuvem, e também examinar a mídia removível que está sendo usada pelo telefone celular, devendo, a depender da situação, fornecer subsídios à autoridade que coordena a investigação para solicitar mais exames.

Ainda segundo Jansen e Ayers, o analisa pericial e o investigador devem buscar informações sobre os envolvidos (quem), determinar a natureza dos eventos (o que), buscar a cronologia dos eventos (quando), determinar a motivação (por que) e como o ato delituoso ocorreu, similarmente ao que ocorre em perícias de computadores.

Dada a grande variedade de telefones celulares e sistemas operacionais voltados para estes tipos de dispositivos, vê-se então que é necessário descrever métodos de analise forense específicos, que serão utilizados a depender da marca, modelo e sistema operacional do dispositivo. Ante o exposto, é necessário que o examinador se especialize em específicos tipos

de telefones celulares, uma vez que o método utilizado em uma situação pode não ser adequado em outra (Ashcroft, 2001).

2.4. O RELATÓRIO (LAUDO)

O relatório deve ser um documento sucinto que contenha todas as informações importantes do processo de apreensão, aquisição e exame. Considerações técnicas a respeito dos procedimentos e métodos utilizados no processo de forense do telefone celular devem ser documentadas e estar com uma linguagem clara e compreensível mesmo por pessoas que não sejam da área de tecnologia. O examinador relator deve estar atento que o judiciário é composto por indivíduos que normalmente não possuem grande afinidade com a área de tecnologia, devendo usar um linguajar simples e direto no relatório, sem deixar de esclarecer os procedimentos utilizados.

O momento para se redigir o relatório é quando o examinador já esgotou todas as possibilidades de interpretar os dados extraídos do telefone celular e já possui as conclusões pertinentes que devem ser agora documentadas de forma clara, objetiva e conclusiva. Todo o recurso disponível deve ser utilizado pelo examinador para poder passar aos leitores as informações obtidas no decorrer dos exames. Figuras, tabelas, anexos, mídias óticas com vídeos extraídos, devem ser utilizados para evitar que informações importantes, evidenciadas pelo examinador durante as outras etapas, não deixem de ser documentadas (Jansen e Ayers, 2007). O examinador deve ter em mente que o resultado de todo o seu trabalho é o relatório.

O relatório deve conter, além dos procedimentos e resultados obtidos, informações claras a respeito do caso, descrição do dispositivo examinado, identificação do examinador, identificação do solicitante dos exames, a data de recebimento da solicitação, a data de emissão do relatório, e uma conclusão clara e concisa a respeito daquilo que foi solicitado.

3. A PLATAFORMA ANDROID

Na Figura 3.1 é apresentada a evolução dos telefones celulares que, com o passar dos anos, demonstra a crescente necessidade dos usuários em agregar cada vez mais informação nesta tecnologia, o que leva ao momento que se vivencia atualmente com o advento dos *smartphones*.

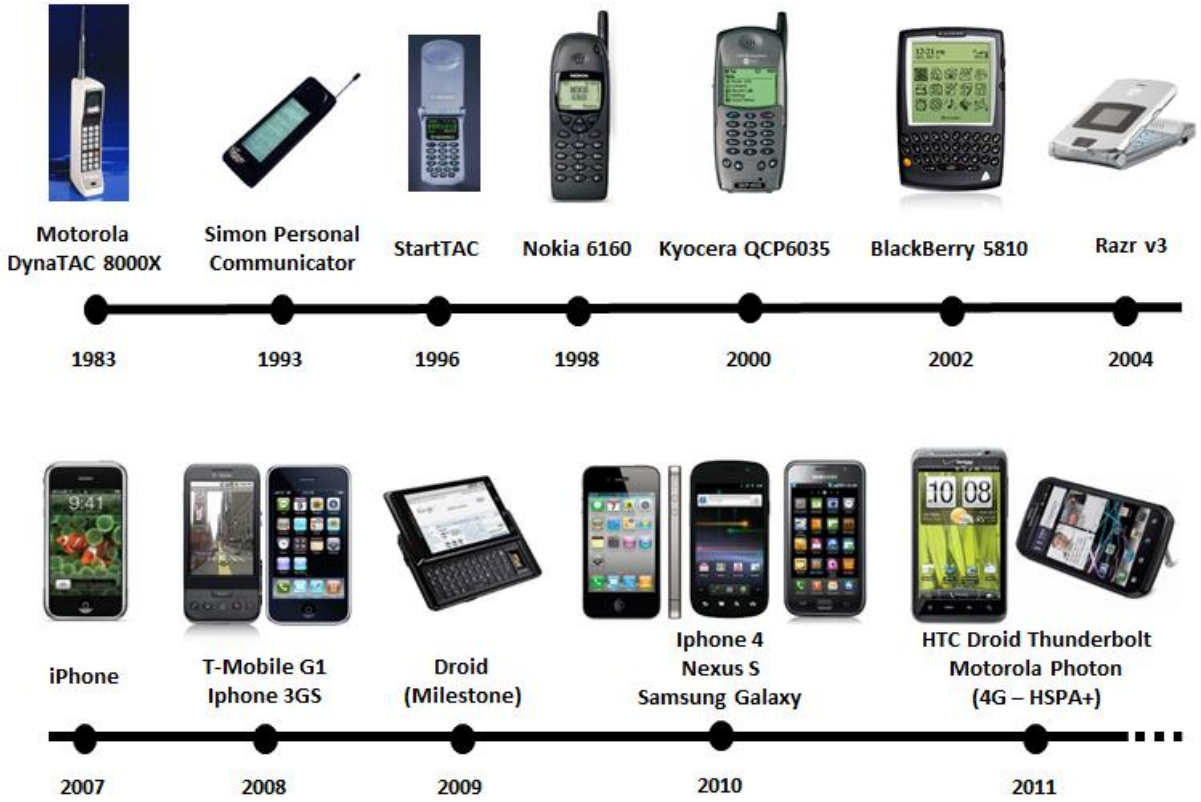


Figura 3.1 - Evolução dos telefones celulares

3.1. BREVE HISTÓRICO DOS TELEFONES CELULARES

O primeiro protótipo de telefone celular foi apresentado pela Motorola em 1973. Foi desenvolvido por Martin Cooper (Farley, 2005). Se chamava DynaTAC 8000X, possuía aproximadamente 30 centímetros, pesava quase 1 quilo e custava US$ 3.995,00, colocado a venda comercialmente dez anos mais tarde, em 1983, quando já possuía bateria suficiente para uma hora de conversação e memória para armazenar 30 números telefônicos (Cassavo, 2007).

Em 1982, a empresa Nokia apresentou o primeiro telefone comercial. Seu nome era “Mobira Senator” e pesava 8,6 quilos a mais que o seu concorrente da Motorola, lançado em 1983

(Speckmann, 2008). Apesar de ser lançado antes do telefone da Motorola, o seu uso era mais voltado para automóveis uma vez que era muito pesado (Cassavo, 2007).

Em 1993, a IBM (*Bell South*) apresentou o primeiro telefone celular com características de PDA (*Personal Data Assistant*) (Cassavo, 2007), o "Simon Personal", que possuía funções de *pager*, calculadora, calendário, assim como fax e e-mail. Pesava em torno de 600 gramas e era vendido por US$ 899,00.

Em 1996, a Motorola apresentou o seu modelo "StarTac", com apenas 87 gramas, que foi um sucesso por apresentar funcionalidades desejadas pelos usuários (agenda, registro e identificador de chamadas) e ser um aparelho com uma estética diferenciada, o que agradou o mercado (Cassavo, 2007).

A Nokia obteve sucesso com os celulares com um desenho diferenciado (*candybar-style*) no final de década de 90, com o lançamento do celular modelo 6160, no ano de 1998, com 170 gramas, e 8260, no ano de 2000, com 96 gramas (Cassavo, 2007). Esta linha de celulares já apresentava funções como calendário e despertador, com cabos de comunicação proprietários e comunicação infravermelha.

O primeiro telefone celular a possuir o sistema operacional Palm foi introduzido pela Kyocera no ano de 2000, no modelo "QCP6035". Já em 2002, foi apresentado pela empresa americana *Danger Hiptop* (T-*Mobile Sidekick*), um dos primeiros aparelhos celulares com navegador web, mensagem eletrônica e mensageria instantânea (Speckmann, 2008).

Com toda a evolução da tecnologia de telefonia celular, com melhorias na rede e aumento do poder computacional, foram lançados vários celulares com as características dos telefones celulares convencionais da atualidade. Em 2002, a empresa *Research In Motion* (RIM), lançou o "BlackBerry 5810", com funcionalidades de mensageria eletrônica, organizador pessoal, calendário e teclado físico. Em 2003, a Nokia lançou o "N-Gage", que era um aparelho celular que também funcionava como um videogame portátil. Já a Motorola investiu no design dos aparelhos celulares, obtendo ainda mais sucesso no seu fino telefone celular "RAZR v3" em 2004, popularizando o telefone celular, uma vez que era desejado por pessoas que possuíam diferentes utilidades para o uso do aparelho (Cassavo, 2007). Já a RIM lançou em 2006 seu primeiro equipamento com câmera embutida e tocador de áudio e vídeo.

Em 2007, a empresa Apple provocou uma grande revolução nos telefones celulares ao apresentar o modelo "iPhone". Aparelho que possuía um grande poder computacional, portabilidade e design, apresentando aos padrões atuais os chamados *smartphones*.

Em 2008, a *Open Handset Alliance* (OHA) lançou oficialmente o sistema operacional para dispositivos móveis Android. Foi uma reposta das empresas líderes do mercado de telefonia celular, juntamente com a empresa Google, ao "iPhone" da Apple, apresentando uma plataforma tão funcional como a do concorrente, entretanto baseada em um sistema aberto e, consequentemente, mais barato. O primeiro celular com o sistema operacional Android a ser comercializado foi o "HTC T-Mobile G1" em 2008.

Ainda em 2008, a Apple lançou seu novo modelo, o "iPhone 3GS", com GPS e conectividade 3G integrados no aparelho. Em 2009 a Motorola lançou o seu modelo "Droid" (equivalente ao modelo Milestone no Brasil) que fez frente ao modelo da Apple, apresentando uma plataforma com um hardware tão bom quanto o do concorrente.

Em 2010, foram lançados os modelos "iPhone 4", "Nexus S" e "Samsung Galaxy S", com melhorias de desempenho, design e interface com o usuário. Neste ano nota-se que o mercado de *smartphones* conseguiu acompanhar o desenvolvimento da até então líder de mercado, Apple, ao lançar dispositivos com funcionalidades tão similares no mesmo período.

Em 2011, são disponibilizados no mercado os primeiros dispositivos móveis Android com a tecnologia móvel de acesso a Internet 4G e HSPA+ (*High Speed Packet Access*), enquanto ainda é aguardado o lançamento do novo telefone da Apple "iPhone 5".

Já as empresas como Microsoft e Nokia, aquela sem sucesso em adotar sua plataforma Windows Phone (antigo Windows CE) e essa perdendo mercado uma vez que seu sistema operacional apresentou limitações na evolução das funcionalidades disponíveis nos dispositivos móveis, se uniram em 2011 para tentar ganhar mais espaço no mercado dos *smartphones* pertencente à Apple e ao Android (Cavaleiro, 2011).

3.2. OS SMARTPHONES COM ANDROID

O Android é um sistema operacional aberto desenvolvido para uso em dispositivos móveis. A empresa mundialmente conhecida, Google Inc. comprou a Android Inc. em 2005, contratando Andy Rubin como diretor do grupo de plataformas móveis (Gadhavi, 2010). Em 5 de

novembro de 2007, a *Open Handset Alliance* (OHA), que é um consórcio de mais de 80 grandes empresas do mercado de dispositivos móveis, como por exemplo, Motorola, Samsung, Sony Ericsson e LG, foi fundada e tem investido e colaborado para o desenvolvimento da plataforma Android. O código fonte do Android foi disponibilizado sob a licença Apache, versão 2.0 de janeiro de 2004[4].

A plataforma Android é composta basicamente pelo sistema operacional, o SDK (*Software Development Kit*) e as aplicações. O SDK é de fácil acesso e utilização, com recursos já bem incorporados no mercado de desenvolvimento de softwares. As aplicações Android usam a linguagem de programação Java, que é bem difundida e aceita. Por questões que serão discutidas *a posteriori*, a Google optou por não utilizar a plataforma Java padrão, sendo escolhida a máquina virtual Dalvik (DVM - Dalvik *Virtual Machine*) em contrapartida.

Uma vez que todo o hardware envolvendo os telefones celulares tende a utilizar memórias flash nos sistemas embarcados, o Android utiliza o sistema de arquivos YAFFS2 (*Yet Another Flash File System* 2), na partição principal no sistema, já que é um sistema de arquivos aberto, voltado para memórias flash e suas peculiaridades, possuindo um desempenho aceitável. Entretanto, a partir do final de 2010 o sistema de arquivos EXT (*extended file system*) começou a aparecer nos celulares com Android (Hoog, 2011), o que significa uma migração para este sistema de arquivos. Esta mudança se justifica pela falta de suporte do YAFFS2 a multitarefa, uma vez que os processadores estão cada vez mais rápidos, inclusive com núcleos múltiplos, assim como pela adoção das memórias eMMC (*Embedded MultiMediaCard*) que trabalham simulando dispositivos de armazenamento em bloco. Cabe também a observação que os sistemas de arquivos citados são os utilizados no Android disponibilizado pela Google. Algumas empresas optam por realizar mudanças, a exemplo do aparelho Samsung Galaxy S9000 que possui o sistema de arquivos RFS (*Robust FAT File System*) em seu Android 2.1.

Conforme ilustra a Figura 3.2, a pilha de software é dividida em quatro camadas, incluindo cinco grupos diferentes:

1) A camada de aplicação: a plataforma Android vem com um conjunto básico de aplicações, a exemplo do navegador web, cliente de mensagem eletrônica, programa para SMS, calendário, contatos, serviço de mapas, dentre outros. Todo o

[4] Pode ser visualizada no sítio da intenet http://www.apache.org/licenses.

sistema é multitarefa, o que permite ao usuário, por exemplo, ouvir música enquanto navega na Internet, sendo as aplicações todas escritas em Java (Google Inc, 2011f).

2) O *framework* de aplicações: é um *framework* de desenvolvimento padronizado e aberto que permite, com a ajuda de provedores de conteúdo e outros serviços, a reutilização das funções das aplicações e seus recursos. Todas as API (*Application Program Interface*) disponíveis para o sistema principal também estão disponíveis para o desenvolvimento das aplicações, o que fornece ao desenvolvedor todos os recursos disponíveis do ambiente (Google Inc, 2011f).
3) As bibliotecas: são escritas em C/C++ e chamadas através de uma interface Java. As funcionalidades disponibilizadas pelas bibliotecas são acessadas através do *framework* de aplicações. Dentre as bibliotecas, estão presentes as que gerenciam as janelas (*surface manager*), gráficos 2D e 3D, media (*codecs*), base de dados SQLite e a WebKit para o navegador web (usada no Google Chrome e Apple Safari) (Hashimi, Komatineni e Maclean, 2010).
4) O ambiente de execução (*runtime*): O ambiente de execução do Android possui um conjunto de bibliotecas que fornece todas as funcionalidades disponíveis nas bibliotecas Java referente à versão do sistema operacional. Estas bibliotecas aumentam suas funcionalidades à medida que as versões do Android são lançadas. A máquina virtual Dalvik trabalha interpretando o código Java e traduzindo para uma linguagem compreensível pelo SO. Ela foi desenvolvida para poder executar múltiplas máquinas virtuais de forma eficiente que, rodando executáveis no formado Dalvik *executable* (“.dex”), consegue otimizar o uso da memória (Ehringer, 2008).
5) O *kernel*: o *kernel* 2.6 do Linux é utilizado pelo sistema operacional Android. Age como uma camada de abstração entre o hardware e a pilha de software, sendo responsável pela gerência de processos no dispositivo (*drivers model*), gerenciamento da memória, gerenciamento da rede e segurança do sistema (Burnette, 2008).

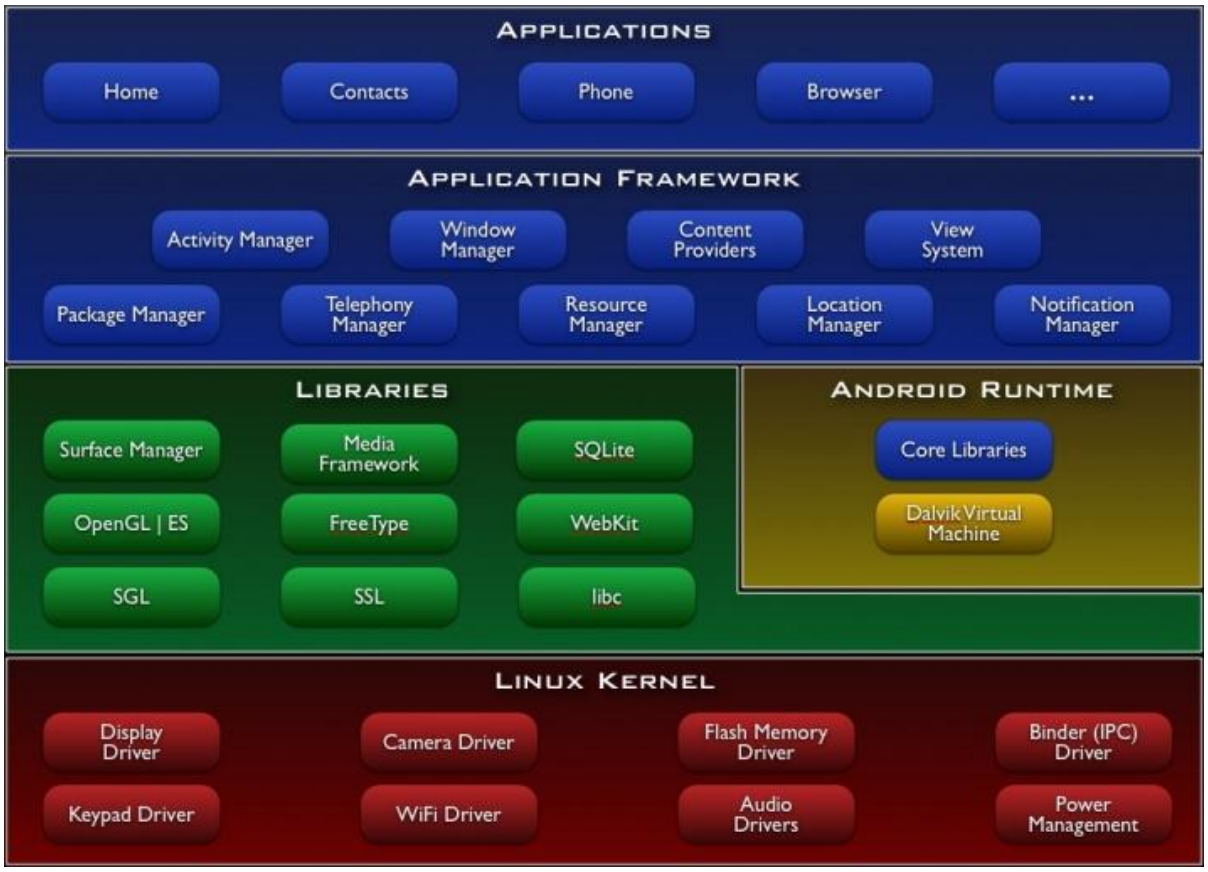


Figura 3.2 - As camadas que compõem o sistema Android (Google Inc, 2011f).

O sistema operacional Android foi escrito para prover ao usuário uma série de funcionalidades padrões que são incorporadas no próprio sistema. A seguir, são listadas as funcionalidades disponibilizadas pelo Android (Google Inc, 2011f):

- *Framework* de aplicações que permite a reutilização de componentes;
- A máquina virtual Dalvik que é otimizada para dispositivos móveis;
- Navegador Web integrado baseado na plataforma aberta WebKit[5];
- Gráficos otimizados utilizando tecnologia 2D customizada e 3D baseado na especificação OpenGL ES 1.0 (aceleração por hardware opcional);
- SQLite para banco de dados relacional;
- Suporte a mídia para formatos padrões de áudio, vídeo e imagem;
- Telefonia GSM[6];
- Bluetooth, 3G, 4G, EDGE e WiFi[6];
- Câmera, GPS, bússola e acelerômetro[6];

[5] http://www.webkit.org/
[6] Depende diretamente do hardware prover os recursos.

- Ambiente de desenvolvimento com emulador de dispositivo, ferramentas de depuração, análise de consumo de memória e desempenho, e um *plugin* para o Eclipse IDE[7].

3.3. VERSÕES DO ANDROID

A Android Inc. era uma pequena empresa localizada na Califórnia, EUA, fundada por Andy Rubin e Rich Miner. O objetivo da empresa era desenvolver um sistema operacional para dispositivos móveis (Bahareth, 2010).

Em 2005, a Google Inc. adquiriu a Android Inc. e manteve os principais funcionários da empresa, a exemplo dos seus fundadores Andy Rubin e Rich Miner. A aquisição deixou claro a intenção da Google em entrar para o mercado de dispositivos móveis.

Em 2007, foi fundada a *Open Handset Alliance*, composta de várias companhias que trabalhavam no mercado de dispositivos móveis, dentre elas a Google Inc., com a finalidade de desenvolver padrões abertos para estes tipos de equipamentos. Desta forma, a adoção do Android como plataforma principal para estabelecer tais padrões, ampliou os investimentos na plataforma.

A Tabela 3.1 mostra as versões da plataforma Android e uma breve descrição de suas funcionalidades associadas (Google Inc, 2011c).

Tabela 3.1 - As versões da plataforma Android.

Versão	Funcionalidades
1.0	Disponibilizada no primeiro *smartphone* com Android, o T-Mobile G1, da HTC, que foi colocado a venda em 22 de outubro de 2008. Já possuía as funcionalidades: alarme, navegador web, calculadora, câmera, contatos, discador, e-mail, mapas, mensageria, músicas, imagens, Android *Market* e configurações.
1.1	Lançada em fevereiro de 2009. Corrigiu vários problemas encontrados na versão anterior, adicionando algumas pequenas funcionalidades às funções já existentes. Foi uma atualização da versão 1.0 apenas para o celular T-Mobile G1 da Google.
1.5 (*Cupcake*)	Lançada em maio de 2009, foram incluídas novas funcionalidades ao sistema. Foi a primeira versão que teve uma boa aceitação comercial.

[7] Ambiente de desenvolvimento integrado que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

	Teve melhorias na interface com o usuário e no desempenho. Algumas das funcionalidades adicionadas foram: • *Kernel* versão 2.6.27; • Gravação e visualização de vídeos; • *Upload* de vídeos para o Youtube e imagens para o Picasa; • Um novo teclado com sugestão de palavras na digitação (*text-prediction*); • Bluetooth A2DP (*Advanced Audio Distribution Profile*) e AVRCP (*Audio/Video Remote Control Profile*); • Conectar automaticamente a dispositivo Bluetooth; • Novos *widgets* e pastas que podiam ser colocadas na tela principal (*home screen*); • Transição animada de tela.
1.6 (*Donut*)	O SDK foi lançado em 15 de setembro de 2009. Algumas das funcionalidades incluídas foram: • *Kernel* versão 2.6.29; • Melhorias no sistema de busca; • Integração da câmera, galeria de fotos e gravação de vídeos; • Atualizações no Android *Market*; • Suporte a reconhecimento de voz convertendo em texto (*text-to-speech*); • Suporte às tecnologias VPN (802.1x) e CDMA;
2.0 / 2.1 (*Eclair*)	O SDK foi lançado em novembro de 2009. Sistema baseado no *kernel* 2.6.29. Apresentou uma interface com usuário renovada e melhora na velocidade do sistema. Outras melhorias implementadas foram: • Suporte a várias resoluções e tamanhos de telas; • Suporte a HTML5 com nova interface do navegador; • Nova lista de contatos; • Maior taxa de contraste para melhorar visualização da tela; • Suporte a sincronização ao Exchange Server; • Possibilidade de usar flash integrado na câmera; • Zoom digital, macro, balanço de branco e modos de cena; • Interface do teclado remodelada; • Suporte a *multi-touch* no teclado; • Bluetooth 2.1;
2.2 (*Froyo*)	O SDK foi lançado em maio de 2010. Sua última versão é a 2.2.2. As mudanças incluíram: • *Kernel* versão 2.6.32; • Mudança na tela inicial do sistema (*home screen*); • Melhorias na segurança do dispositivo, com a utilização de senhas com caracteres alfanuméricos e possibilidade de restaurar configurações de fábrica remotamente; • Compartilhamento da rede 3G pela rede Wi-Fi (3G *hotspot*) e pela USB (USB *tethering*); • Aperfeiçoamento no consumo de memória e energia e do

	desempenho do sistema operacional; • Utilização da compilação JIT (*Just In Time*), que melhorou o desempenho das aplicações JAVA; • Suporte a instalação de aplicações no cartão de memória; • Suporte ao Adobe Flash Player 10.1+; • Suporte a resoluções de tela com maior definição.
2.3 (*Gingerbread*)	O SDK foi lançado em dezembro de 2010. Apresentou novas funcionalidades para o desenvolvedor, ao usuário com inserção de novas tecnologias. Sua última versão é a 2.3.3. As mudanças incluíram: • *Kernel* versão 2.6.35; • Modificada a interface com o usuário e a entrada de texto; • Melhoria no suporte a telas maiores com melhores resoluções (WXGA e superior); • Suporte nativo a telefonia SIP VoIP; • Melhorias no suporte a vídeos e áudio; • Melhorias nas funcionalidades de desenvolvimento de jogos; • Melhorias no desempenho do *garbage collection* (limpeza das áreas apagadas do sistema de armazenamento – memória flash) e *drivers* de vídeo; • Suporte a mais sensores a exemplo do giroscópio e barômetros; • Melhoria no gerenciamento de memória e das aplicações; • Suporte nativo a múltiplas câmeras e NFC (*Near Field Communication*);
3.0 / 3.1 / 3.2 (*Honeycomb*)	O SDK foi lançado em fevereiro de 2011, voltado exclusivamente para *tablets*. As mudanças incluíram: • Otimização do sistema para utilização em *tablets*; • Área de trabalho em três dimensões com *widgets* redesenhados e tela inicial customizável; • Sistema multitarefa refinado; • Melhorias no navegador como a utilização de guias (*tabs*), autopreenchimento, sincronização de favoritos com o Google Chrome e navegação privada; • Aceleração por hardware de gráficos 2D; • Suporte a processadores com vários núcleos.

3.4. APLICATIVOS DA PLATAFORMA ANDROID

Todas as aplicações que rodam no sistema operacional Android são escritas na linguagem de programação Java. A Google fornece ao desenvolvedor ferramentas que compõem o SDK (*Software Development Kit*), a fim de fornecer um ambiente para realizar a programação, compilação e disponibilização do pacote de instalação.

Os aplicativos são compilados com todas as referências, recursos e dados em um arquivo com a extensão ".apk", denominado *Android Package*, que é utilizado para realizar a instalação do programa no sistema operacional (Hashimi, Komatineni e Maclean, 2010).

Como o sistema é multiusuário, é possível cada aplicação ser executada com seu próprio usuário. A fim de restringir o acesso aos arquivos de cada aplicação, o sistema utiliza um ID de usuário único para cada programa, que é usado para configurar as permissões nos arquivos referentes à aplicação. Ademais, cada processo roda em uma máquina virtual independente, assim o código em execução é isolado do acesso das demais aplicações. Cada aplicação possui seu processo, que é inicializado quando é solicitado pelo sistema, liberando o espaço reservado em memória quando o processo não é mais requerido (Google Inc, 2011b).

Depois de realizada a instalação do aplicativo no sistema operacional, cada programa possui sua área de segurança reservada (*sandbox*). O sistema Android consegue isolar o ambiente de execução de cada aplicação e restringe o acesso aos arquivos instalados. Nesta configuração, a aplicação não pode acessar áreas que não são permitidas (Google Inc, 2011b).

Apesar de fornecer uma área de isolamento adequada, em algumas situações, as aplicações desejam trocar informações entre si e utilizar funcionalidades e serviços umas das outras. Neste contexto, é possível fazer com que duas aplicações compartilhem o mesmo ID de usuário, o que permitiria realizar o compartilhamento dos arquivos. Utilizando o mesmo ID, é possível às aplicações rodarem no mesmo processo e compartilharem a mesma máquina virtual (Google Inc, 2011b).

Outra forma de conseguir acessar os recursos de outras aplicações é no momento da instalação do aplicativo, a partir das permissões definidas no arquivo de configuração *AndroidManifest.xml*. Na sua instalação, o aplicativo pode solicitar ao usuário que ele dê permissões de acesso aos demais serviços instalados no sistema, a exemplo da lista de contatos, GPS, cartão de memória, câmera e bluetooth. Uma vez que o usuário autoriza o acesso aos recursos, a aplicação poderá fazer chamadas a fim de utilizá-los durante sua execução (Google Inc, 2011b).

Neste arquivo, o desenvolvedor da aplicação deve declarar todos os componentes da sua aplicação. Desta forma, é possível ao sistema conhecer todos os componentes disponibilizados pelo programa. Ademais, é utilizado para definir o nível de API (API *level*)

requerido pela aplicação, que está vinculado a compatibilidade com as versões do Android (Google Inc, 2011b).

Dentre outras funções, o arquivo *manifest* é utilizado para requerer ao usuário permissões de acesso a recursos instalados no telefone celular, como por exemplo, ler os contatos ou realizar escrita no cartão SD e até mesmo os recursos de hardware, como acesso a câmera, bluetooth ou à tela *multitouch*.

3.5. O ANDROID *MARKET*

Com a finalidade de disponibilizar aos usuários do sistema Android um repositório de aplicações desenvolvidas especialmente para a plataforma, a Google desenvolveu e disponibilizou uma ferramenta que vem pré-instalada em todos os dispositivos, denominada Android *Market*.

Por meio do Android *Market*, aplicativo gratuito fornecido pela Google, o usuário do sistema pode buscar aplicações desenvolvidas por terceiros e instalá-las no dispositivo, sejam elas pagas ou sem custo. Além disso, gerencia todas as compras realizadas pelo usuário, vinculando os aplicativos adquiridos a sua conta Google. Também mantém o sistema atualizado, fornecendo ao usuário informações a respeito da atualização de versões dos aplicativos já baixados.

Para que uma empresa ou um indivíduo possa disponibilizar uma aplicação no Android *Market*, ele deve ter uma conta Google, criar uma conta de distribuidor (*publisher*), pagar uma taxa e concordar com os termos de uso (Hashimi, Komatineni e Maclean, 2010).

3.6. A ARQUITETURA DA MÁQUINA VIRTUAL DALVIK

A plataforma Java foi criada pela Sun em 1995. Desde sua publicação, seu objetivo era provar uma plataforma de desenvolvimento e execução com aplicativos portáveis que pudessem ser utilizados em vários ambientes de execução, ou seja, “escreva uma vez, rode em qualquer lugar” (Ehringer, 2008).

A Google adotou o Java como plataforma de desenvolvimento e execução do seu sistema operacional para dispositivos móveis, o Android. Entretanto, optou por não utilizar a tecnologia Java padrão, optando pela Dalvik *Virtual Machine* (DMV), com uma implementação limitada das bibliotecas padrões.

Em ambiente de desktop, Java *Standard Edition* - JSE, e de servidor, Java *Enterprise Edition* - JEE, a plataforma Java conseguiu atingir o seu objetivo. Entretanto, a implantação em ambiente portátil de dispositivos pessoais móveis não teve a mesma receptividade. Apesar de ter sido lançado o JME (Java *Micro Edition*), com o foco do mercado de dispositivos móveis, oferecendo uma plataforma de desenvolvimento mais reduzida (dada a limitação de hardware imposta), ele teve baixa aceitação.

A utilização de uma tecnologia não padronizada pode trazer consequências futuras que acabam a enfraquecendo. Afinal, a utilização daquilo que não é condizente com o padrão de mercado acaba dividindo a tecnologia, o que leva, no caso do Java, a mudar o foco inicial da sua criação, que seria sua fácil portabilidade (Ehringer, 2008). Entretanto, o pouco sucesso do Java da Sun entre os desenvolvedores de tecnologias móveis, se dá devido à falta de uma plataforma que pudesse oferecer um desempenho aceitável em hardware de baixo desempenho.

A Máquina Virtual Dalvik conseguiu utilizar várias funcionalidades da JVM, assim como de todas as bibliotecas padrões Java, e ainda sim prover customizações adequadas ao ambiente móvel (Ehringer, 2008), motivo pelo qual a Google a adotou em seu sistema operacional móvel.

A plataforma Android foi concebida para funcionar exclusivamente em dispositivos móveis, a exemplo de telefones celulares e *tablet* PCs. Alguns requisitos mínimos de hardware foram descritos e mudam dependendo da versão do Android a ser instalado no equipamento (Google Inc, 2011d).

Dada a variedade de dispositivos móveis que podem suportar o sistema operacional Android, a portabilidade da camada de aplicação, com abstração do hardware, é vital para a existência do sistema. Para poder manter a estabilidade e segurança do sistema Android, dada esta diversidade de aplicações que são desenvolvidas por diferentes desenvolvedores, é importante que os programas rodem em ambiente controlado, segregado e independente, de tal forma que não influenciem na robustez do sistema e na estabilidade dos demais aplicativos.

Partindo da premissa da necessidade de se ter um sistema que possa ter um ambiente de execução com limitada velocidade de processador, sem espaço para paginação, alimentado por bateria, memória RAM limitada, grande variedade de dispositivos, e aplicações que rodem em espaço independente e reservado, foi necessário escolher a Máquina Virtual

Dalvik, pois consome pouca memória, consegue isolar as aplicações em tempo de execução e usa a linguagem bem difundida Java.

Cada aplicação Android roda seu próprio processo, com sua própria instância da máquina virtual Dalvik. A DVM foi desenvolvida para que cada dispositivo rode múltiplas máquinas virtuais eficientemente. A fim de diminuir o consumo de memória, a DVM executa arquivos no formato ".dex" (*Dalvik Executable*). É uma máquina virtual *register-based*[8] que roda classes compiladas por um compilador Java que foram transformadas no formato ".dex" por uma ferramenta chamada "dx" (Ehringer, 2008).

3.6.1. O formato ".dex"

No ambiente Java convencional, o *bytecode* Java é produto da compilação de um código fonte Java e é armazenado nos arquivos ".class". Para cada classe referenciada dentro do código fonte Java, será criado um arquivo ".class".

Na plataforma Android que utiliza da máquina virtual Dalvik, os arquivos ".class" gerados a partir de um código fonte Java passam pela ferramenta "dx". Esta ferramenta faz com que sua saída, um arquivo ".dex", contenha as múltiplas classes utilizadas no código fonte original, sendo este o executável DVM (Ehringer, 2008).

A ferramenta "dx" realiza este processo com a finalidade de otimizar o uso da memória por meio do compartilhamento dos dados. O formato ".dex" utiliza um *pool* de constantes e tipos específicos como seu mecanismo primário para conservar a memória (Ehringer, 2008). Assim, as constantes que antes eram armazenadas cada uma em sua classe específica agora são agrupadas em um único *pool* quando os arquivos ".class" são submetidos à ferramenta "dx", que elimina a duplicidade de constantes no processo de criação do executável ".dex" (Hashimi, Komatineni e Maclean, 2010).

A máquina virtual Dalvik, consegue por meio deste processo de criação do seu executável ".dex", diminuir consideravelmente o consumo de memória, uma vez que, segundo Denis Antonioli e Markus Pilz (Antonioli e Pilz, 2008), a maior parte de uma classe Java está no seu

[8] Segundo estudos apresentados pelo Institute of Management Sciences (Khan, Khan, *et al.*, 2009), a arquitetura *register-based* requer 47% menos instruções do que uma arquitetura *stack-based* (predominante no mercado). O código daquela é 25% maior, mas este aumento custa apenas um extra de 1,07% dos recursos de uma VM, sendo sua performance 32,3% melhor do que uma *stack-based*.

pool de constantes (62%) e não na parte do método que representa apenas 33% do tamanho do arquivo. As outras partes do arquivo compartilham apenas 5% do seu tamanho.

No artigo de David Ehringer (Ehringer, 2008), é exposta a Tabela 3.2 (Bornstein, 2008), que demonstra o considerável ganho de memória no uso dos executáveis da DVM.

Tabela 3.2 - Comparação dos tamanhos dos arquivos Java.

Código	Arquivo JAR sem compressão (bytes)	Arquivo JAR com compressão (bytes)	Arquivo DEX sem compressão (bytes)
Bibliotecas do sistema	21.445.320 (100%)	10.662.048 (50%)	10.311.972 (48%)
App de navegador web	470.312 (100%)	232.065 (49%)	209.248 (44%)
App de alarme e relógio	119.200 (100%)	61.658 (52%)	53.020 (44%)

Outro fator importante na máquina virtual Dalvik é o processo de coleta de lixo da memória (*garbage collection*) (Bornstein, 2008). A estratégia de coleta de lixo deve preservar o compartilhamento da memória, quando um dado objeto não deve ser apagado se tiver sendo usado por mais de um processo. Entretanto, na DVM cada aplicação roda um processo separado e assim como o sistema de coleta de lixo, onde cada aplicação tem o seu. A solução para evitar que uma coleta de lixo de um processo não limpe um objeto que está sendo compartilhado em memória foi a de utilizar bits de marcação, que indica que um determinado objeto é "alcançável" e, portanto, não pode ser excluído da memória.

Para resolver o problema de lentidão na inicialização dos aplicativos Java, assim como do compartilhamento de código que é utilizado em vários aplicativos, o Android usa o conceito chamado Zigoto. O Zigoto assume que há um número significante de classes java e estruturas que são usadas por vários aplicativos (Ehringer, 2008). Assume que estas estruturas são apenas de leitura, ou seja, jamais poderão ser alteradas. Assim, consegue se aperfeiçoar o compartilhamento da memória entre os processos.

Assim que o sistema Android inicializa, ainda durante o processo de boot, um processo chamado Zigoto é inicializado. Este processo carrega algumas classes de bibliotecas primárias. Estas classes de bibliotecas primárias servem para fornecer a um processo um "pré-carregamento" e compartilhamento, sendo apenas para leitura (Brady, 2008).

Assim que inicializa, o processo Zigoto fica em um estado de espera aguardando receber uma solicitação em seu *socket* para realizar um "fork" de uma nova instância baseada na sua

máquina virtual. Assim é possível já obter todas as classes de bibliotecas primárias já carregadas em uma nova VM que rodará em um processo independente, isolado e consequentemente seguro.

As bibliotecas primárias já pré carregadas a partir do "fork" do processo Zigoto são todas protegidas contra escrita. Elas só serão modificadas quando o processo Zigoto que originou os demais processos sofrer alguma alteração em suas bibliotecas primárias, sendo que os dados alterados no processo originário serão copiados imediatamente para suas instâncias. Desta forma, consegue-se prover uma rápida atualização dos dados compartilhados e proteção contra a alteração indevida destas bibliotecas primárias por outros processos (Bornstein, 2008). Diferente das máquinas virtuais Java tradicionais, que não possuem este tipo de compartilhamento de bibliotecas, onde cada instância receberá uma cópia inteira das bibliotecas primárias e dos objetos associados, ou seja, a memória não é compartilhada pelas instâncias (Bornstein, 2008).

3.7. A PLATAFORMA ANDROID SOB A ÓTICA PERICIAL

Nesta seção, serão apresentados alguns princípios básicos de forense no sistema Android, a fim de subsidiar o método proposto no capítulo 4, apresentando características específicas e ferramentas que o analista pericial poderá utilizar no processo forense. Antes de realizar uma análise forense de um *smartphone* com o sistema Android, deve-se ter alguns conceitos a respeito da plataforma, suas ferramentas e funcionalidades. Assim, o analista pericial terá o conhecimento necessário sobre os recursos que poderá utilizar e, juntamente com sua expertise, realizar o exame no dispositivo.

3.7.1. O SDK do Android

O SDK do Android é um conjunto de ferramentas que possuem como objetivo fornecer aos desenvolvedores da plataforma um ambiente completo para criação e depuração dos aplicativos Android. O aplicativo principal é o *SDK Manager*, onde é possível baixar as APIs referentes às diferentes versões do Android e executar emuladores do sistema. O SDK também disponibiliza algumas outras ferramentas, a exemplo da ferramenta "dx" usada para gerar o arquivo executável Dalvik e da ADB (*Android Debug Bridge*), usada para se conectar a um emulador ou dispositivo Android em modo de depuração USB e realizar algumas ações no sistema via linha de comando.

O ADB provê uma interface ao dispositivo Android conectado ao computador ou a um emulador Android gerenciado pelo *SDK Manager*. Normalmente encontra-se instalado no diretório *<sdk>/plataform-tools*.

É uma ferramenta que trabalha na arquitetura cliente-servidor com três componentes (Google Inc, 2011a):

1) Cliente: é utilizado por um terminal ou linha de comando através da ferramenta ADB na máquina à qual o dispositivo está conectado.
2) Servidor: também fica em execução na máquina à qual o dispositivo está conectado. É executado em segundo plano como um serviço e gerencia a comunicação entre o cliente o serviço (*daemon*) que está em execução no dispositivo.
3) Serviço (*daemon*): é executado em segundo plano no dispositivo.

O servidor é inicializado quando um cliente faz uma chamada para realizar uma conexão a um dispositivo Android (Google Inc, 2011a). A partir do momento que o servidor consegue estabelecer uma conexão com o serviço no dispositivo ou emulador, comandos ADB podem ser utilizados para gerenciar o dispositivo. Diferentemente de conexões realizadas a emuladores Android, só é possível realizar uma conexão com o serviço em um dispositivo físico se estiver habilitada a opção "Depuração USB" nas suas configurações. Caso a depuração USB não esteja habilitada, não é possível realizar conexão ADB com o dispositivo.

Por meio ADB, via do modo de depuração, é possível conectar ao dispositivo Android e obter um *shell*. Pode-se instalar aplicativos, copiar arquivos, obter informações do sistema e obter informações de log (*logcat*). A Figura 3.3 possui alguns dos comandos que podem ser executados a partir da ferramenta ADB. Por meio de um *shell* no dispositivo, podem ser executados comandos nativos do ambiente GNU/Linux diretamente no dispositivo ou emulador. Alguns exemplos de comandos que podem ser executados no *shell* são: *ls*, *cd*, *rmdir*, *mkdir*, *cp*, *rm*, *cat*, *pwd*. Entretanto é um *shell* básico, sem comandos mais avançados. Por padrão, a conexão via ADB em um dispositivo físico é realizada com o usuário "shell" (uid=2000), que é muito restritivo; nas conexões feitas para um emulador a permissão é de super usuário (*root*). Para se ter o acesso a um *shell* com permissões de super usuário em um dispositivo físico, é preciso que o sistema esteja com acesso à *root* instalado, que será explicado adiante.

```
Android Debug Bridge version 1.0.26

-d                            - directs command to the only connected USB device
returns an error if more than one USB device is present.
-e                            - directs command to the only running emulator.
returns an error if more than one emulator is running.
-s <serial number>            - directs command to the USB device or emulator with
the given serial number. Overrides ANDROID_SERIAL
environment variable.

device commands:
adb push <local> <remote>    - copy file/dir to device
adb pull <remote> [<local>]  - copy file/dir from device
adb shell                    - run remote shell interactively
adb shell <command>          - run remote shell command
adb logcat [ <filter-spec> ] - View device log
adb install [-l] [-r] [-s] <file> - push this package file to the device and install it
('-l' means forward-lock the app)
('-r' means reinstall the app, keeping its data)
('-s' means install on SD card instead of internal storage)
adb uninstall [-k] <package> - remove this app package from the device
adb help                     - show this help message
adb version                  - show version num
```

Figura 3.3 - Ajuda da ferramenta ADB com lista de parâmetros da ferramenta.

Por meio do *shell*, é possível executar comandos como o *dmesg*, que imprime na tela informações de debug do Kernel do Linux; o *dumpsys* retorna informações sobre o sistema e os aplicativos[9]. Por meio deste comando, pode-se obter informações do estado do Wi-Fi, das janelas do sistema, da CPU, da memória, das atividades, dos processos, dentre outras (Paula, 2011).

O Android também fornece em seu *shell* o *logcat* (vide Figura 3.4). Por meio do *logcat* é possível visualizar e filtrar as mensagens de debug do sistema e de aplicações, armazenadas pelo próprio sistema que utiliza buffers circulares (Google Inc, 2011e).

```
Usage: logcat [options] [filterspecs]
options include:
-s              Set default filter to silent.
Like specifying filterspec '*:s'
-f <filename>   Log to file. Default to stdout
-r [<kbytes>]   Rotate log every kbytes. (16 if unspecified). Requires
-f
-n <count>      Sets max number of rotated logs to <count>, default 4
-v <format>     Sets the log print format, where <format> is one of:
brief process tag thread raw time threadtime long
-c              clear (flush) the entire log and exit
-d              dump the log and then exit (don't block)
-t <count>      print only the most recent <count> lines (implies -d)
-g              get the size of the log's ring buffer and exit
-b <buffer>     request alternate ring buffer
('main' (default), 'radio', 'events')
-B              output the log in binary
filterspecs are a series of
<tag>[:priority]

where <tag> is a log component tag (or * for all) and priority is:
V    Verbose
D    Debug
I    Info
W    Warn
E    Error
F    Fatal
```

[9] O comando “dumpsys | grep DUMP” retorma alguns dos parâmetros específicos usados no dumsys.

```
S    Silent (supress all output)

'*' means '*:d' and <tag> by itself means <tag>:v

If not specified on the commandline, filterspec is set from
ANDROID_LOG_TAGS.
If no filterspec is found, filter defaults to '*:I'

If not specified with -v, format is set from ANDROID_PRINTF_LOG
or defaults to "brief"
```

Figura 3.4 - Comandos e filtros disponíveis no *logcat*.

3.7.2. Estrutura do sistema de arquivos do Android

Os celulares com sistema Android utilizam memória flash. A fim de que a memória flash possa ser tratada de forma convencional pelo sistema, é necessário haver uma camada de firmware chamada FTL (*Flash Translation Layer*), com a finalidade de, juntamente com o subsistema MTD (*Memory Technology Device*), permitir ao sistema trabalhar com a memória como se ela fosse um dispositivo de blocos convencional, a exemplo de um disco rígido, funcionado como um tradutor de requisições (linux-mtd.infradead.org, 2008).

Desta forma, os pontos de montagem da estrutura de todo o sistema operacional Android é por meio do MTD. Para obter uma lista de todos os pontos de montagem do sistema, basta executar o comando conforme demonstrado na Figura 3.5 (Hoog, 2009).

```
$ adb -s <dispostivo> shell cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00180000 00020000 "pds"
mtd1: 00060000 00020000 "cid"
mtd2: 00060000 00020000 "misc"
mtd3: 00380000 00020000 "boot"
mtd4: 00480000 00020000 "recovery"
mtd5: 008c0000 00020000 "cdrom"
mtd6: 0afa0000 00020000 "system"
mtd7: 06a00000 00020000 "cache"
mtd8: 0c520000 00020000 "userdata"
mtd9: 00180000 00020000 "cust"
mtd10: 00200000 00020000 "kpanic"
```

Figura 3.5 - Pontos de montagem do sistema Android[10].

Também é possível obter a lista de todos os pontos de montagem através do comando demonstrado na Figura 3.6.

```
$ adb -s <dispostivo> shell mount
rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,relatime,mode=755 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
none /acct cgroup rw,relatime,cpuacct 0 0
tmpfs /mnt/asec tmpfs rw,relatime,mode=755,gid=1000 0 0
/dev/block/mtdblock6 /system yaffs2 ro,relatime 0 0
/dev/block/mtdblock8 /data yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock7 /cache yaffs2 rw,nosuid,nodev,relatime 0 0
```

[10] O sistema cria os dispositivos "/dev/mtd/mtdX" e "/dev/mtd/mtdXro".

```
tmpfs /dev tmpfs rw,relatime,mode=755 0 0
devpts /dev/pts devpts rw,relatime,mode=600 0 0
tmpfs /mnt/asec tmpfs rw,relatime,mode=755,gid=1000 0 0
none /dev/cpuctl cgroup rw,relatime,cpu 0 0
/dev/block/mtdblock5 /cdrom yaffs2 rw,relatime 0 0
tmpfs /tmp tmpfs rw,relatime,size=2048k 0 0
/dev/block/mtdblock0 /pds yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/vold/179:1 /mnt/sdcard vfat rw,dirsync,nosuid,nodev,noexec,relatime,u
id=1000,gid=1015,fmask=0702,dmask=0702,allow_utime=0020,codepage=cp437,iocharset
=iso8859-1,shortname=mixed,utf8,errors=remount-ro 0 0
/dev/block/vold/179:1 /mnt/secure/asec vfat rw,dirsync,nosuid,nodev,noexec,relat
ime,uid=1000,gid=1015,fmask=0702,dmask=0702,allow_utime=0020,codepage=cp437,ioch
arset=iso8859-1,shortname=mixed,utf8,errors=remount-ro 0 0
tmpfs /mnt/sdcard/.android_secure tmpfs ro,relatime,size=0k,mode=000 0 0
/dev/block/dm-0 /mnt/asec/uk.co.nickfines.RealCalc-1 vfat ro,dirsync,nosuid,node
v,noexec,relatime,uid=1000,fmask=0222,dmask=0222,codepage=cp437,iocharset=iso885
9-1,shortname=mixed,utf8,errors=remount-ro 0 0
/dev/block/dm-1 /mnt/asec/com.halfbrick.fruitninja-1 vfat ro,dirsync,nosuid,node
v,noexec,relatime,uid=1000,fmask=0222,dmask=0222,codepage=cp437,iocharset=iso885
9-1,shortname=mixed,utf8,errors=remount-ro 0 0
/dev/block/dm-2 /mnt/asec/com.ArtInGames.AirAttackHDLite-1 vfat ro,dirsync,nosui
d,nodev,noexec,relatime,uid=1000,fmask=0222,dmask=0222,codepage=cp437,iocharset=
iso8859-1,shortname=mixed,utf8,errors=remount-ro 0 0
/dev/block/dm-3 /mnt/asec/com.reigndesign.Pigrush-1 vfat ro,dirsync,nosuid,nodev
,noexec,relatime,uid=1000,fmask=0222,dmask=0222,codepage=cp437,iocharset=iso8859
-1,shortname=mixed,utf8,errors=remount-ro 0 0
/dev/block/dm-4 /mnt/asec/zok.android.dots-1 vfat ro,dirsync,nosuid,nodev,noexec
,relatime,uid=1000,fmask=0222,dmask=0222,codepage=cp437,iocharset=iso8859-1,shor
tname=mixed,utf8,errors=remount-ro 0 0
```

Figura 3.6 - Comando *mount* do *shell* do Android.

Pode-se observar a partir da saída dos comandos *cat /proc/mtd* e *mount* que as partições definidas para o sistema, para os dados do usuário e para o *cache* são respectivamente os diretórios */system*, */data* e */cache* (Hoog, 2009). Nota-se também que os aplicativos instalados no cartão de memória possuem pontos de montagem específicos. Assim é possível obter facilmente quais os aplicativos encontram-se instalados no cartão de memória[11].

Obtendo diretamente um *shell* do sistema, é possível navegar pela estrutura do sistema de arquivos. Na Figura 3.7 é ilustrado os diretórios e arquivos listados em um *shell* de um celular com o sistema Android versão 2.2.2 instalado.

```
$ pwd
/
$ ls -l
lrwxrwxrwx root     root              2011-04-04 11:32 config -> /pds
drwxrwxrwt root     root              2011-04-04 11:32 tmp
drwxrwxr-x system   system            2011-02-09 19:40 pds
drwxr-xr-x system   system            2010-02-11 19:00 cdrom
-rw-r--r-- root     root        14543 2011-04-04 11:32 init.rc
-rw-r--r-- root     root        16976 2011-04-04 11:32 init.mapphone_umts.rc
drwxrwx--x system   cache             2011-04-04 23:11 cache
lrwxrwxrwx root     root              2011-04-04 11:32 sdcard -> /mnt/sdcard
drwxr-xr-x root     root              2011-04-04 11:32 acct
drwxrwxr-x root     system            2011-04-04 11:32 mnt
lrwxrwxrwx root     root              2011-04-04 11:32 d -> /sys/kernel/debug
lrwxrwxrwx root     root              2011-04-04 11:32 etc -> /system/etc
drwxr-xr-x root     root              2011-04-03 16:24 system
drwxr-xr-x root     root              1969-12-31 21:00 sys
drwxr-x--- root     root              1969-12-31 21:00 sbin
```

[11] Só é possível instalar nativamente aplicativos no cartão removível a partir da versão 2.2 do Android.

```
dr-xr-xr-x root     root              1969-12-31 21:00 proc
-rwxr-x--- root     root            453 1969-12-31 21:00 init_prep_keypad.sh
-rwxr-x--- root     root           6840 1969-12-31 21:00 init.mapphone_cdma.rc
-rwxr-x--- root     root           1677 1969-12-31 21:00 init.goldfish.rc
-rwxr-x--- root     root         108632 1969-12-31 21:00 init
-rw-r--r-- root     root            118 1969-12-31 21:00 default.prop
drwxrwx--x system   system              2011-04-04 11:32 data
drwx------ root     root              2011-03-05 06:17 root
drwxr-xr-x root     root              2011-04-04 11:32 dev
```

Figura 3.7 - Comandos executados diretamente no *shell* de um Android versão 2.2.2.

A Tabela 3.3 descreve os principais diretórios do sistema apresentando uma breve descrição de cada um (Hoog, 2009).

Tabela 3.3 - Principais diretórios dos dados do usuário do sistema Android.

Diretório		Descrição
/data/	dalvik-*cache*/	Arquivos Dalvik *Executable* (.dex) das aplicações Java.
	anr/	Abreviação para *Application Not Responding*. Possui informações de debug com os *timestamps* dos arquivos de log gerados.
	app/	Arquivos .apk de aplicações instaladas.
	data/	Armazena as configurações e base de dados SQLite das aplicações. Cada aplicação tem sua estrutura de diretórios.
	misc/	Configurações do usuário de DHCP, Wi-Fi, bluetooth, VPN, dentre outras.
	system/	Armazena informações sobre configurações do sistema relativas às aplicações e seus usos. Lista dos aplicativos em *packages.list* ou *packages.xml*.

3.7.3. SQLite Database

A plataforma Android optou por utilizar o banco de dados SQLite para prover às aplicações um gerenciador de banco de dados relacional, leve, robusto e de simples utilização. O SQLite foi criado por Richard Hipp em 2000. É um banco de dados que não necessita de configurações. É utilizado pela Apple, Nokia, Mozilla Firefox, Skype, Adobe, Solaris, etc (Burnette, 2008). Não há restrições para o uso do banco de dados, sendo de domínio público, com código aberto.

Utilizando uma biblioteca compacta, que, segundo informações obtidas do sítio do SQLite, possui no máximo 300 KiB, ocupando apenas 4 KiB na pilha de execução, tornou-se uma plataforma muito utilizada em dispositivos com recursos de hardware limitados, a exemplos dos *smartphones*. O SQLite não possui um servidor associado, com um serviço dedicado

sendo executado em segundo plano. Simplesmente faz leitura e escrita diretamente no sistema de arquivos, possuindo uma estrutura completa de um banco de dados (tabelas, *views*, *triggers*, índices) em um único arquivo (SQLite, 2011).

No sistema Android, os arquivos com a extensão ".db" se referem à bancos de dados do tipo SQLite. Cada aplicação só possui acesso ao seu banco de dados e, caso tenha que utilizar um banco de outra aplicação, em tempo de execução fará uma solicitação ao sistema que, verificando as devidas permissões realizará a consulta mostrando os resultados.

Para se acessar o banco de dados, basta que o sistema operacional forneça as permissões ao arquivo. Não há um controle de acesso definido pelo SQLite. Para o acesso a uma base SQLite, a segurança reside nas permissões do sistema.

3.7.4. Permissões de super usuário (root) no sistema Android

O sistema operacional Android foi concebido para ser um sistema leve e robusto. Com a finalidade de evitar que o proprietário do dispositivo realize interações com o sistema que possam afetar a estabilidade e confiabilidade do Android. Os usuários usados por padrão pelos aplicativos, não possuem permissões que realizem modificações no sistema no Android. Esses usuários são bem restritivos e apenas interagem com permissões de realizar as atividades específicas para aquele aplicativo, sem que modificar estruturas mais elaboradas do sistema operacional.

Entretanto, proprietários mais avançados dos celulares Android, viram a necessidade de obter um maior controle sobre o sistema operacional. Através de técnicas que buscam vulnerabilidades no próprio sistema ou acesso aos *bootloaders*[12] dos *smartphones*, é possível obter permissões de super usuário (*root*). As técnicas para se obter o acesso de super usuário no Android variam conforme fabricante, modelo do telefone celular e versão do sistema, sendo muitas vezes invasivas, podendo inclusive danificar os dados armazenados no aparelho.

A partir do acesso de super usuário, o proprietário do dispositivo pode, por meio de um *shell*, executar comandos com perfil de *root*, podendo realizar qualquer tarefa dentro do sistema operacional, a exemplo de realizar *overclocks*, backups de aplicativos restritos, acessar diretórios das partições do sistema. Desta forma, alguns aplicativos também podem ser

[12] Programas especialmente construído para que seja capaz de carregar outro programa que permite a iniciação do sistema operacional (Hallinan, 2010)

executados com perfil de *root*, não estando mais limitados as permissões convencionais, onde o próprio sistema "blindava" a aplicação para que ela não acessasse os dados das demais aplicações.

Sob o ponto de vista do analista pericial, um celular Android com permissões de acesso de super usuário é muito mais interessante. Isso porque poderá realizar operações no dispositivo que antes não eram possíveis, a exemplo de acessar partições de sistema e dados do usuário através de um *shell* fornecido pela ferramenta ADB. Com permissões de super usuário, o analista pericial pode espelhar todas as partições disponíveis no sistema[13] para uma análise *post mortem*, sem maiores intervenções no equipamento.

[13] Conforme citado por Hoog (Hoog, 2009), a fim de espelhar as partições do telefone, deve fazer referências aos dispositivos */dev/mtd/mtdX* e */dev/mtd/mtdXro*.

4. O MÉTODO PROPOSTO

Este capítulo é integralmente dedicado a descrever o método proposto para realizar uma análise pericial em um *smartphone* com sistema operacional Android. O método é baseado nas melhores práticas utilizadas atualmente pela Polícia Federal do Brasil, pelo NIST (Jansen e Ayers, 2007), pelo Departamento de Justiça dos Estados Unidos (Ashcroft, 2001), polícia Inglesa (Association of Chief Police Officers, 2008) e Instituto Forense da Holanda (Netherlands Forensic Institute, 2007), e busca diferenciar-se adaptando essas melhores práticas às peculiaridades da plataforma Android.

Telefones celulares com o sistema operacional Android oferecem muitas funcionalidades, como navegar na Internet, armazenar imagens e vídeos, documentos, calendário, contatos, localização GPS, mapas, dentre outras. A facilidade para desenvolver, publicar e instalar aplicativos no sistema Android faz com que a plataforma possua funcionalidades que só serão limitadas pela capacidade do desenvolvedor, enriquecendo as funcionalidades do *smartphone*.

Devido a esta grande capacidade de prover diversas funcionalidades, sob a ótica forense, um celular com o sistema Android pode armazenar grande quantidade de informação sobre o seu usuário, sendo uma excelente "testemunha" para provar fatos, ou obter informações para uma investigação (Rossi, 2008).

Entretanto, uma análise pericial em celulares difere da abordagem utilizada em computadores pessoais. A aquisição da memória interna de um aparelho celular, inclusive daqueles com o sistema Android, é um processo mais complexo do que a abordagem de se retirar um disco rígido do computador e espelhá-lo (copiá-lo). Em celulares, é utilizada uma memória interna para instalação do sistema operacional e suas principais funcionalidades, sendo sua remoção e cópia, um procedimento mais complexo do que a de um disposto de memória não-volátil como um disco rígido, um cartão de memória ou um disco de estado sólido (*Solid State Disk* - SSD).

A abordagem de aquisição e exame dos dados, utilizada em ambientes de computadores convencionais, onde as informações contidas na memória podem ser preservadas na sua totalidade, não se aplica em ambientes de *smartphones* e celulares; dada as dificuldades expostas anteriormente, somada às peculiaridades, a exemplo de diferentes hardwares, que os fabricantes disponibilizam e utilizam em seus dispositivos.

O sistema Android possui mecanismos de autenticação que podem ser ativados ou não pelo usuário. Dentre estes mecanismos, podemos citar a tela bloqueada por senha ou por padrão táctil, onde o acesso ao sistema só ocorre se for colocada a sequência correta de autenticação. Ademais, a depender da versão e customização do Android, a integração com uma conta da Google, por exemplo, pode ser maior, onde o desbloqueio pode ocorrer através da inserção da senha correta do usuário ou por meio da biometria (impressão datiloscópica).

Além das especificidades descritas, como o Android é um sistema aberto, existe aplicativos que podem ser instalados que "desbloqueiam" o usuário do sistema. É possível executar aplicativos e comandos com poder de super usuário (*root*), o que torna o sistema acessível na sua integralidade, apesar de tal procedimento não ser recomendável pelos fabricantes dos equipamentos celulares e pela Google, uma vez que o sistema pode ser totalmente modificado pelo seu usuário e os aplicativos podem realizar qualquer função dentro do sistema, o que fere princípios de segurança.

O método proposto objetiva mostrar ao analista pericial como proceder no momento da apreensão, aquisição dos dados, exame e documentação (laudo/relatório) de *smartphones* com o sistema Android. A partir do conhecimento obtido a partir dos capítulos anteriores, é possível utilizar o método descrito, com a finalidade de poder extrair o máximo de informações do dispositivo, de forma a resguardar e documentar adequadamente a evidência que está sendo processada.

A Figura 4.1 ilustra o fluxograma geral no tratamento de evidências digitais a exemplo de telefones celulares e computadores. A partir desse fluxograma, é possível visualizar as etapas que os analistas periciais, ou equipes investigativas se depararão do momento em que o *smartphone* é encontrado até o encaminhamento do relatório/laudo pericial, quando o trabalho do especialista é finalizado. O método proposto para análise pericial em *smartphones* com sistema Android especificará as etapas de "a. Apreensão", "e. Aquisição dos dados", "f. Exame" e, consequentemente, "g. Geração do Relatório/Laudo Pericial".

Os subprocessos "a. Apreensão", "e. Aquisição de dados" e "f. Exame" (vide Figura 4.1) são os de maior complexidade e envolvem outros processos e decisões mais específicos, e serão esclarecidos nas seções 4.1, 4.2 e 4.3, respectivamente. Os demais processos fazem parte da atividade administrativa da polícia e serão descritas a seguir a fim de fornecer um entendimento basal ao leitor sobre o processamento de uma evidência.

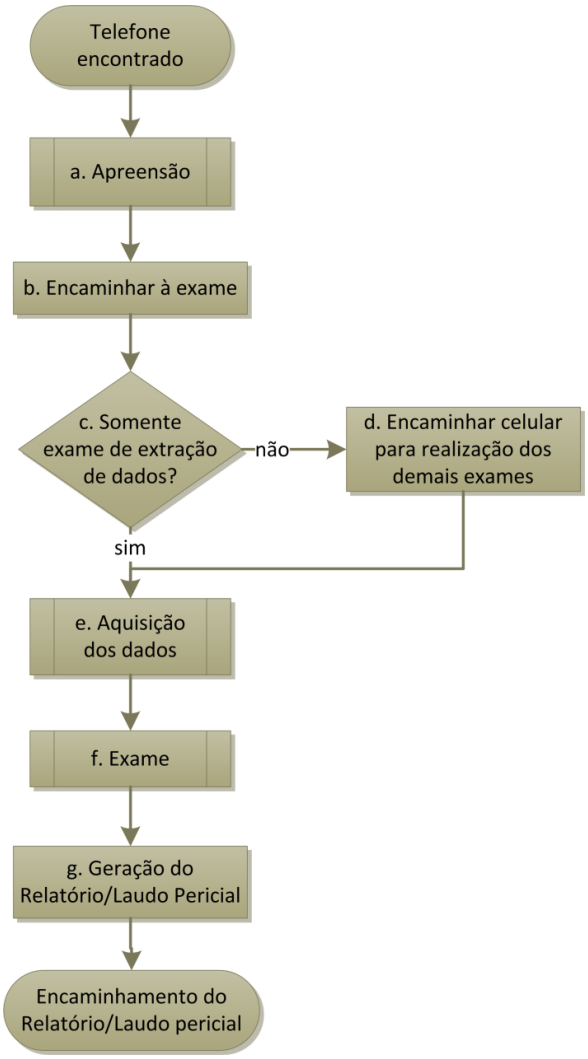


Figura 4.1 – Fluxograma geral com as etapas do método proposto

O processo “b. Encaminhar à exame” é um procedimento comum no processo investigativo brasileiro e é de responsabilidade da autoridade responsável pela investigação criminal, ou seja, o Delegado de Polícia ou do Ministério Público Federal. Geralmente, quando há um material a ser examinado para produção de prova material, ele é encaminhado aos Peritos Criminais, para que estes especialistas realizem o exame necessário na evidência a fim de materializar a prova, contribuindo de forma única para determinação de autoria e *modus operandi* do delito em questão. A autoridade solicitante dos exames se atentará para o fato que a evidência, no caso o telefone celular, pode ser objeto de exames como DNA, impressões papilares, valoração, podendo até mesmo ser urgente por conta da vida da bateria ou quando o suspeito está preso, e realiza a solicitação dos exames periciais de acordo com o que está sendo apurado.

No processo “c. Somente exame de extração de dados”, a secretaria administrativa do setor de perícias, ou até mesmo o analista pericial designado para realização dos exames, deve verificar, no momento do recebimento da solicitação de pericia, se é necessária a realização de outros exames além dos dados armazenados no dispositivo (“c. Somente exame de extração de dados?”). Havendo a necessidade de demais exames (“d. Encaminhar celular para realização de demais exames”), estes devem preceder o de extração, uma vez que o manuseamento do equipamento pode destruir evidências que poderiam ser coletadas. Entretanto, os exames que serão realizados antes da extração também devem ser conduzidos de forma a não prejudicar a aquisição dos dados, a exemplo da utilização de produtos de revelação de impressões papilares que podem prejudicar o funcionamento do telefone celular, podendo ser discutido qual procedimento pericial deverá ser realizado prioritariamente.

Os processos “e. Aquisição dos dados”, “f. Exame” e “g. Geração do Relatório/Laudo Pericial” são exclusivos do especialista da área. Apenas o perito pode manusear o *smartphone* nos processos de aquisição de dados e exame, realizando a devida documentação com a finalidade de descrever todo o trabalho realizado no relatório/laudo pericial de forma clara, objetiva e conclusiva.

Os procedimentos adotados durante os processos descritos no fluxograma da Figura 4.1 são todos formalizados a partir de algum documento. O resultado do processo “a. Apreensão” é um Auto de Apreensão (Brasil, 2003), lavrado por um Escrivão de Polícia e, a depender da situação, também é elaborado em um Laudo Pericial de Local de Crime, redigido por um Perito Criminal. O processo “b. Encaminhar à exame” é um memorando ou ofício do Delegado de Polícia, ou do membro do Ministério Público ao Perito Chefe do Setor de Perícias. Os processos “c. Somente exame de extração de dados?” e “d. Encaminhar celular para realização dos demais exames” são documentados a partir de despachos e designações de exames periciais pelo Chefe do Setor de Perícias. O processo “e. Aquisição dos dados” e “f. Exame” são as anotações, realizadas pelo Perito designado para realização da análise pericial, e os relatórios gerados pelas ferramentas utilizadas no decorrer destes processos. Já o processo “g. Geração do Relatório/Laudo pericial” é o documento que compõe todo o trabalho desenvolvido pelo Perito, assim como algumas considerações a partir dos documentos gerados a partir das etapas anteriores, contendo o histórico da apreensão, quando for o caso, assim como a identificação autoridade solicitante dos exames e o objetivo do que está sendo apurado.

4.1. APREENSÃO

O processo de apreensão de um *smartphone* com o sistema Android deve, preferencialmente, ser acompanhado de um analista pericial que tenha conhecimento sobre a plataforma. O fluxograma ilustrado na Figura 4.2 descreve detalhadamente o subprocesso "a. Apreensão" da Figura 4.1.

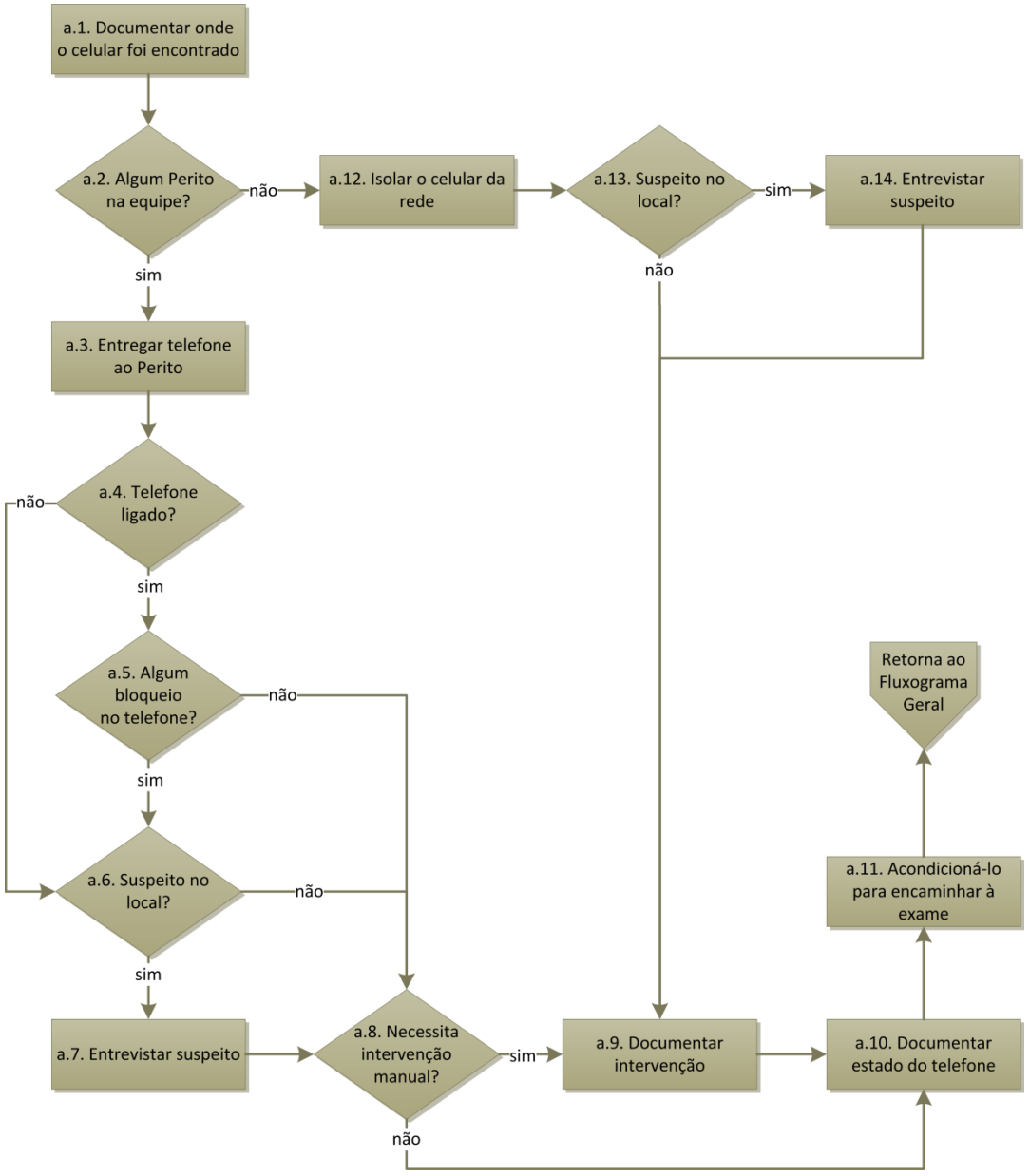


Figura 4.2 - A apreensão de um *smartphone* com o sistema Android.

Ao encontrar o *smartphone*, deve-se documentar onde ele foi encontrado (a.1). Se o dispositivo estiver desligado é interessante a equipe observar se está com a bateria inserida, caso contrário deverá buscá-la. Também é recomendável buscar por cartões de memória

removíveis e cartões SIM, uma vez que podem ser utilizados no telefone. Normalmente, no processo de busca e apreensão, testemunhas acompanham toda a ação da equipe, sendo inclusive essencial para provar a idoneidade desta etapa. Se houver alguma avaria aparente no equipamento, esta deve ser documentada na sua descrição, a fim de resguardar a equipe de apreensão de ser acusada de ter provocado danos intencionalmente. Deve-se também documentar onde o telefone se encontrava. Se estiver em posse de uma pessoa, deve ser descrito no documento de apreensão (auto de apreensão) seu nome e identificação pessoal (RG, CPF, CNH). Se for encontrado em um local, descrever no documento de apreensão quais as pessoas que mais frequentavam àquela região ou cômodo.

Cabe a observação de que, a depender do local ou o motivo da busca, seja necessária a preservação das provas materiais que podem ser contaminadas na apreensão, a exemplo de uma impressão papiloscópica, ou um resto de tecido para realização de exames de DNA. Nestas situações, é importante a equipe atentar para preservação no sentido de utilizar os equipamentos de proteção individual (EPI), como luvas, uniformes e toucas a fim de coletar as evidências.

4.1.1. O papel do analista pericial

Deve-se verificar se na equipe de apreensão há algum analista pericial com conhecimentos sobre a plataforma Android (a.2). Caso um perito esteja presente na equipe, o *smartphone* deve ser entregue a ele (a.3) para que possa dar prosseguimento à apreensão. O perito, com sua expertise, pode tomar as decisões corretas estando o telefone ligado ou não (a.4). No caso de estar ligado, ele pode avaliar se há algum bloqueio de acesso ao dispositivo (a.5), caso não tenha nenhum bloqueio, sequer há a necessidade de se entrevistar o suspeito (que nem sempre passa informações verídicas) e avaliará a necessidade de realizar uma intervenção manual ou não no celular (a.8), a exemplo de isolá-lo da rede ou analisar a suíte de aplicativos instalados no equipamento. Esta análise dos aplicativos deve ir além do menu de aplicativos do dispositivo, devendo o perito observar também no menu de configuração e, a depender da situação, nos aplicativos baixados do Android *Market* (se o telefone não tiver sido isolado da rede). Isso porque alguns aplicativos podem não aparecer no menu de aplicativos, mas estarão presentes no menu de configuração ou no Android *Market*.

Estando o *smartphone* bloqueado, o analista pericial deve entrevistar o suspeito caso esse esteja presente (a.6) com a finalidade de obter os códigos de desbloqueio do celular. Outras

informações como e-mail e senha da conta Google, uso do sistema de geolocalização, expertise do suspeito no sistema Android, podem ser úteis no momento do exame. Toda intervenção realizada no celular deve ser documentada (a.9), assim como o estado do telefone (a.10), se está bloqueado ou não, se o suspeito forneceu a senha ou não, se tem algum aplicativo que possa influenciar no exame, se o aparelho estava desligado, IMEI, número do SIM, etc.

O perito que estiver no local deve realizar intervenções no sentido de buscar informações a respeito do uso da plataforma Android pelo proprietário do *smartphone* (a.8), isolando-o da rede de telefonia no momento mais conveniente. Essa intervenção é justificável em situações em que a não realização desse procedimento possa acarretar em perda de informação ou inviabilização da extração dos dados do *smartphone*, que será realizada em um segundo momento. Inclusive, sempre que o dispositivo estiver ligado, é interessante configurá-lo em modo de avião (*offline*). Ademais, o especialista deve observar, no momento da apreensão, se possuem aplicativos que podem sobrescrever informações importantes armazenadas no *smartphone* como, por exemplo, uma ferramenta de backup e que pode ter uma tarefa agendada para restaurar o sistema em um momento futuro anterior à fase de extração dos dados.

4.1.2. Primando pela preservação

A falta do perito no momento da apreensão não é recomendada, entretanto é possível de ocorrer, seja pela falta de informação sobre o suspeito ou pelo julgamento de não haver a necessidade de sua presença pela autoridade responsável pela busca e apreensão. Se não houver nenhum especialista, a equipe deve isolar (a.12) o *smartphone* da rede de telefonia e de dados (desativar os serviços de 3G, 4G, Wifi, GSM, *bluetooth*), seja por meio do uso de um invólucro que bloqueia os sinais ou colocando-o em modo de avião. Em último caso, se a equipe não tiver um invólucro ou não tiver conhecimento suficiente para colocar o telefone em modo de avião, deve remover a bateria ou desligar o equipamento para isolá-lo da rede. Este procedimento provoca perda da memória volátil do telefone, entretanto, é justificável devido ao sistema Android permitir a instalação de aplicativos que remotamente, via rede de telefonia ou Internet, podem apagar os dados do aparelho, bloqueá-lo com códigos de acesso

ou realizar restaurações de cópias de segurança feitas em um momento anterior[14]. Apenas depois de o aparelho estar isolado da rede e, consequentemente, preservado, a equipe deve entrevistar o suspeito caso ele esteja no local (a.13), tentando obter possíveis códigos de acesso do telefone celular, senhas, endereço de e-mail, usuários de Internet usados no telefone e outros dados mais, a depender da experiência do entrevistador (a.14).

4.1.3. Considerações sobre a apreensão

Todo o processo de apreensão deve ser devidamente documentado (a.9) para posterior descrição do equipamento de forma mais detalhada (a.10), e acondicionamento para encaminhá-lo a exame (a.11). A documentação da apreensão é importante para as fases seguintes, pois é o início da cadeia de custódia e preservação do histórico da evidência. Muitas informações só podem ser obtidas nesta etapa e influenciarão as decisões a serem tomadas nas demais fases, podendo inclusive viabilizar uma perícia se estiver documentado, por exemplo, um código de desbloqueio fornecido pelo suspeito no momento da apreensão, ou a desativação de um aplicativo que apaga informações importantes antes da extração dos dados do sistema.

4.2. AQUISIÇÃO DOS DADOS

A aquisição dos dados é o momento mais técnico, onde a habilidade do perito é importante para realização dos procedimentos de extração da forma apropriada. Este processo é mais complexo e necessita de um especialista com conhecimento específico sobre a plataforma Android, pois poderá haver a necessidade de uma intervenção manual para viabilizar a correta aquisição da informação. O fluxograma ilustrado na Figura 4.3 detalha as decisões e processos que um analista pericial pode se deparar ao realizar a aquisição dos dados de um *smartphone* com o sistema Android.

[14] Seek droid, My Android Protection 2.0, Lookout Mobile Security, Norton, Wheres My Droid, Lost Phone, etc.

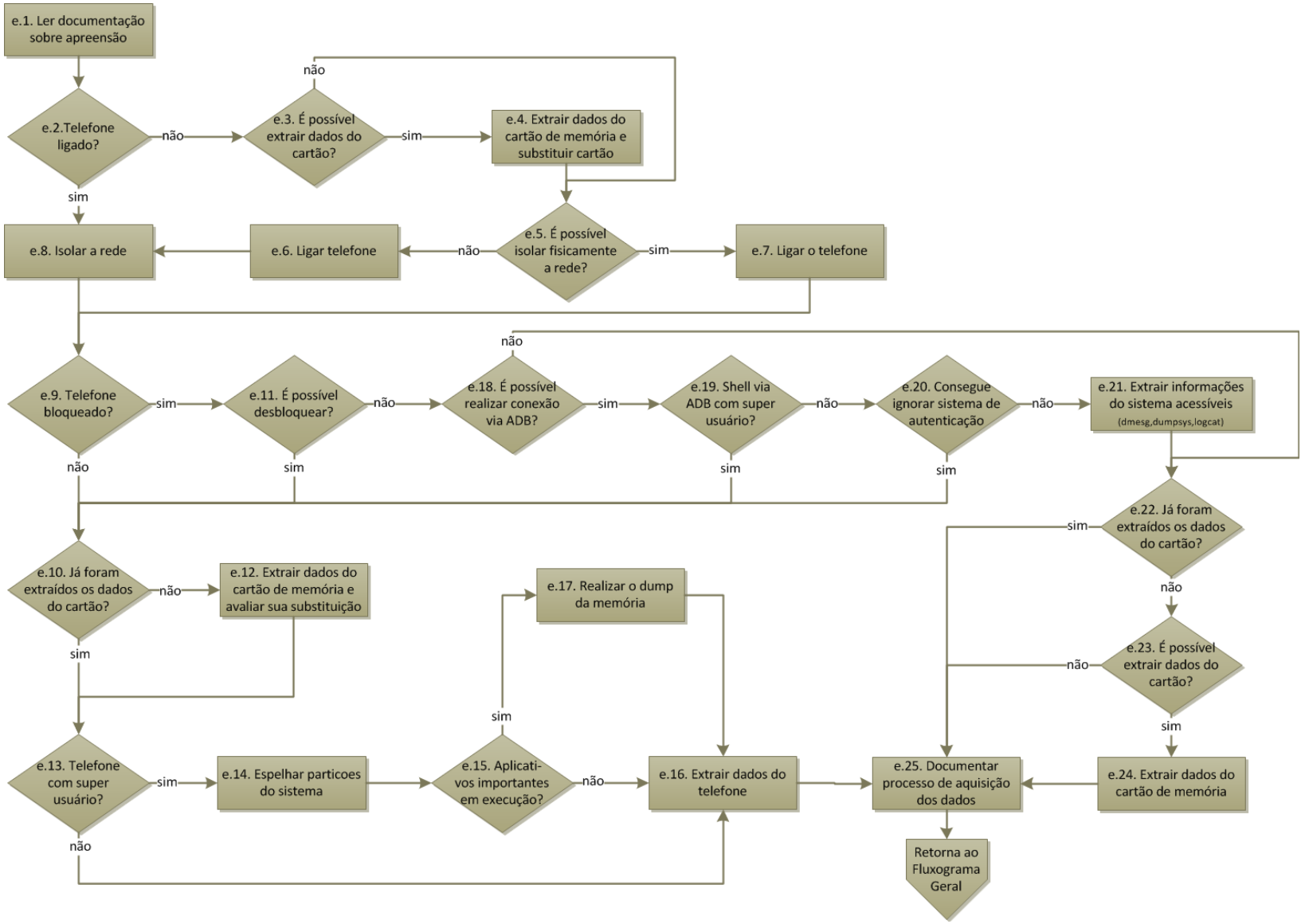


Figura 4.3 - Etapa de aquisição dos dados de um telefone celular com o sistema operacional Android.

Nesta etapa, o perito deve se preocupar em extrair a maior quantidade de informação com o mínimo de intervenção no sistema, que provavelmente será inevitável. Em um primeiro momento (e.1), o analista pericial deverá se inteirar sobre o processo de apreensão (fluxograma da Figura 4.1, processo "a"), lendo a documentação produzida (auto de apreensão e informações técnicas)[15], e se informar a respeito do que está sendo solicitado nos exames (fluxograma da Figura 4.1, processo "b"), a fim de subsidiar as decisões a serem tomadas no processo de extração dos dados do sistema Android.

4.2.1. Aquisição e preservação dos dados do cartão e do *smartphone*

A Figura 4.4 ilustra a etapa inicial na aquisição dos dados do dispositivo Android. Estando o *smartphone* desligado, deve-se preocupar em extrair as informações do cartão de memória que possam ser removidos do dispositivo (e.4), substituindo o cartão original por um cartão de memória do examinador, espelhando aquele para esse. Cabe a ressalva que alguns modelos de celulares Android não podem ter os dados do cartão de memória copiados neste momento, uma vez que são internos e não removíveis (e.3). Para duplicar integralmente os dados do cartão de memória, pode-se utilizar a mesma abordagem utilizada em dispositivos de memória do tipo USB, como os *pendrives*. Para a cópia pode-se utilizar ferramentas forenses e gerar o *hash* dos dados duplicados. Ao término do procedimento, o cartão de memória do examinar, que contém a cópia, deve ser inserido no aparelho.

A próxima preocupação é verificar se é possível isolar a rede de telefonia fisicamente (e.5), por meio de uma sala com isolamento de sinais eletromagnéticos, a fim de evitar a comunicação do telefone com fontes externas antes de ligá-lo (e.7). Caso não seja possível isolar fisicamente a rede, o analista pericial deve ligá-lo (e.6) e imediatamente colocá-lo em modo de avião[16] (e.8). Se o telefone estiver ligado, o analista deve priorizar isolar a rede fisicamente e, em não sendo possível, ativar o modo de avião para dar prosseguimento à extração.

[15] Os documentos produzidos na fase de apreensão devem ser encaminhados junto com o telefone celular apreendido no processo "b. Encaminhar a exame" do fluxograma da Figura 4.1.
[16] Caso o telefone tenha recebido alguma mensagem, ou algum aplicativo ter realizado alguma iteração com a rede de telefonia, o analista deve documentar o fato no seu relatório/laudo pericial.

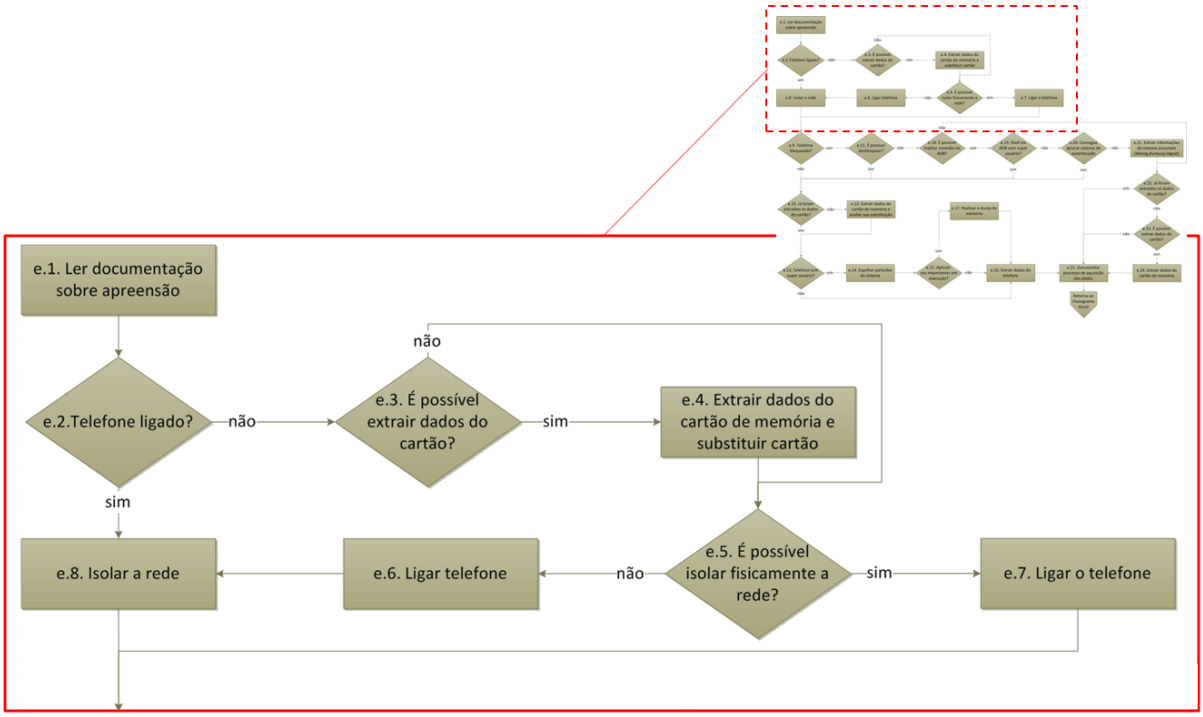


Figura 4.4 - Procedimentos iniciais na aquisição dos dados do dispositivo Android.

4.2.2. A aquisição dos dados de *smartphone* sem controle de acesso

Os processo de aquisição dos dados em dispositivos Android se controle de acesso ativado encontram-se ilustrados na Figura 4.5. Com o telefone ligado, é possível verificar se ele possui algum tipo de controle de acesso ativado. Não estando bloqueado (e.9) ou após desbloqueá-lo (e.11 e e.20) ou ainda se ele tiver acesso de depuração (ADB) com permissões de super usuário (e.19), caso não tenha sido realizada a extração dos dados do(s) cartão(ões) (e.10), este será o momento de extraí-los[17]. Deve-se utilizar bloqueio de escrita, e, se possível, clonar (duplicar integralmente) seu conteúdo para um cartão do examinador que irá substituí-lo no aparelho (e.12). Deve ser avaliada a substituição do cartão original em situações em que não seja possível a sua remoção, ou quando se tem a necessidade de remover a bateria para retirá-lo não sendo desejável o desligamento do *smartphone*. Nestas situações é justificável a utilização do cartão original para realização da perícia após ter sido integralmente copiado, com o analista pericial documentando e justificando o procedimento.

[17] O analista pericial deve se atentar ao fato de que há no mercado aparelhos que possuem um cartão de memória externo, que pode ser removido do dispositivo, e outro interno, que não pode ser removido (eMMC – *Embedded MultiMediaCard*).

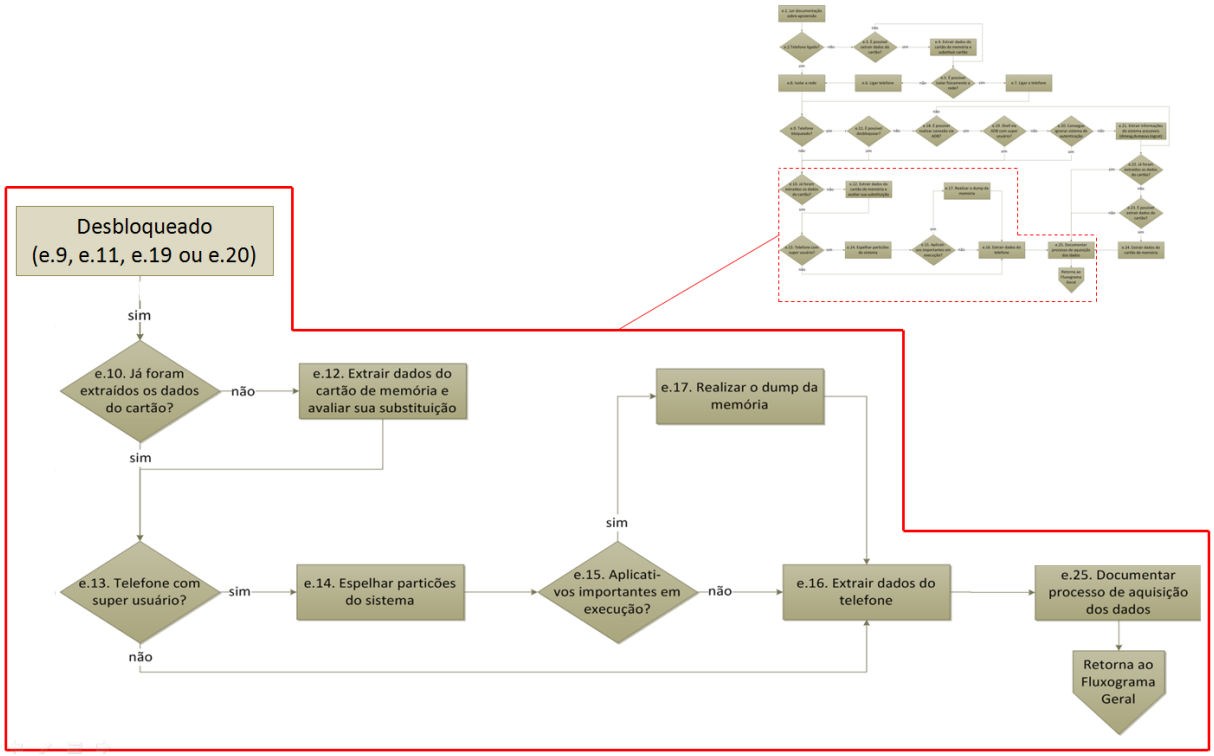


Figura 4.5 – Processos que envolvem um *smartphone* sem controle de acesso.

4.2.2.1. Verificação das permissões de super usuário

Em seguida, é preciso verificar se o aparelho se encontra com permissões de super usuário (e.13). Estando com estas permissões ativadas, deve-se copiar o conteúdo integral das partições do sistema (e.14). O processo de espelhar as partições do sistema com permissões de super usuário é forma mais completa de obtenção dos dados. É possível ao analista pericial espelhar integralmente as partições do sistema por meio dos comandos *cat* ou *dd*. É importante esclarecer que, nas técnicas atuais, os arquivos gerados com o conteúdo da partições do sistema (imagens) serão gravados no cartão de memória instalado no aparelho. Depois, quando for o caso (e.15), deve extrair os dados em memória dos processos que se encontram em execução (e.17), para ter acesso a informações sensíveis (Cannon, 2010), como senhas e chaves criptográficas.

4.2.2.2. Extração dos dados do telefone

Com os dados do cartão de memória original resguardados, assim como do sistema quando for o caso, inicia-se o processo de extração de dados do telefone (e.16). Neste processo o cartão que se encontra no aparelho deve estar devidamente preparado para realizar a extração dos dados do telefone, com espaço suficiente para armazenar os dados que serão extraídos.

Em não havendo espaço suficiente no cartão de memória para extrair as informações necessárias, o analista pericial poderá, se possível, providenciar um cartão de maior capacidade, espelhando o cartão original para este novo cartão. Em último caso, nas situações e quem não foi possível substituir o cartão apreendido junto com o aparelho por um do examinador, pode-se avaliar o que pode ser apagado do cartão de memória para liberar espaço de armazenamento para extração. Todo o procedimento deverá ser documentado com as devidas ressalvas. Outra opção é utilizar as ferramentas para extração forense de dados de telefones celulares disponíveis no mercado, a exemplo da Cellebrite UFED (Cellebrite, 2011).

Caso o sistema tenha permissões de super usuário, nesta fase de extração de dados do telefone (e.16), apesar de já ter sido realizada a cópia integral dos dados do sistema, recomenda-se copiar os arquivos de banco de dados, *cache* das aplicações e os arquivos de configuração do sistema, e até mesmo realizar uma inspeção visual, a fim de facilitar o acesso a estas informações na etapa de exame, assim como arquivos gerados nos processos anteriores a exemplo dos arquivos de *dump* de memória[18]. Isso porque para a análise posterior dos dados extraídos, o analista pericial deve ter um ambiente de exames com ferramentas para montar imagens com suporte ao sistema de arquivos utilizado no dispositivo, geralmente o YAFFS2, que não se encontram disponíveis no mercado. A criação desta redundância poderá ser útil no momento dos exames, principalmente em situações que não seja necessário aprofundar a análise das partições do sistema. Ademais, alguns aplicativos podem estar ativos no sistema, e uma simples inspeção visual pode prover informação que seria de difícil acesso por meio da análise da imagem gerada.

Para realizar a extração dos dados do telefone (e.16) de forma mais amigável e eficiente, é necessária a instalação de um ou mais aplicativos no *smartphone*. Em sendo possível, esta instalação deve ser realizada por meio da ferramenta ADB (vide seção 3.7.1), com o dispositivo configurando para aceitar conexão USB no modo de depuração. Outra forma de instalá-los é por meio de um gerenciador de aplicativos que permita a navegação no cartão de memória, onde eles deverão estar armazenados, com o telefone configurado para aceitar instalações de aplicativos que não são do Android *Market*. Ao instalar um aplicativo forense no telefone celular, os dados do aparelho são modificados. Entretanto, esta abordagem é

[18] O processo de realizar o *dump* de memória de um aplicativo citado por Cannon envolve a necessidade de interromper abruptamente a sua execução e consequente escrita do arquivo de *dump* na memória principal do sistema. Este arquivo de *dump* deve ser copiado para uma estação pericial (por meio da ferramenta ADB) ou para o cartão de memória (caso seja um cartão de memória do examinador).

interessante para se conseguir extrair todos os dados acessíveis ao usuário no telefone, devendo assim ter acesso manual ao telefone. Para isso, o aplicativo forense terá no arquivo "AndroidManifest.xml" (vide seção 3.4) todas as permissões necessárias para realizar a extração.

A fim de complementar a extração dos dados do telefone (e.16), o analista pericial poderá navegar pelo sistema Android sob a perspectiva do seu proprietário, podendo inclusive observar a forma com que ele utilizava os aplicativos instalados, conferindo as últimas ligações efetuadas e recebidas, analisando as mensagens de e-mail e de texto, dentre outras atividades. Inclusive, recomenda-se ao analista pericial comparar os dados obtidos a partir da extração do aplicativo forense com os dados efetivamente armazenados no telefone celular com a finalidade de auditar as informações colhidas e complementá-las quando necessário.

4.2.3. A aquisição dos dados de *smartphone* com controle de acesso

No caso de *smartphone* com restrição de acesso (e.9), em que não foi possível desbloqueá-los (e.11), o perito pode ainda tentar realizar uma conexão USB com a estação pericial e tentar acessá-los via ADB (e.17). Se o acesso via ADB não estiver liberado, não será viável ao analista pericial extrair os dados contidos no sistema do telefone, a não ser que utilize técnicas de extração mais agressivas[19] (que devem ser discutidas na realização do exame), restando ao analista pericial realizar a extração do cartão de memória apreendido juntamente com o telefone, caso ainda não tenha sido realizada (e.22, e.23 e e.24).

Caso contrário, o perito poderá tentar obter um *shell* com permissões de super usuário (e.19), e, em sendo possível, prosseguir com aquisição dos dados. Caso o especialista consiga utilizar a ferramenta ADB, mas não consiga permissões de super usuário, poderá ainda tentar configurar o sistema para ignorar o sistema de autenticação (e.20), instalando aplicativo para este fim (Cannon, 2011)[20]. Na técnica descrita por Cannon, é necessário que a senha da conta Google esteja cadastrada no dispositivo Android, assim como habilitado o acesso à Internet, o

[19] Estas técnicas, que não são o foco deste trabalho, envolveriam o acesso ao *bootloader* do dispositivo, que varia a depender do aparelho e modelo, ou extração da memória física do telefone celular. Uma vez que podem danificar o equipamento, seria necessário entrar em contato com a autoridade solicita para avaliar a necessidade de realização dos exames esclarecendo que o aparelho poderia ser definitivamente danificado, sem possibilidade de recuperação.

[20] "Screen lock Bypass" disponível no Android Market.

que é desaconselhável. Desta forma, recomenda-se obter o aplicativo a partir de outro dispositivo Android e instalá-lo via ADB no dispositivo móvel examinado.

Em não sendo viável ignorar o sistema de autenticação, ainda é possível obter informações das ferramentas de log do sistema (e.21), a exemplo do *dmesg*, *dumpsys* e *logcat*, uma vez que, a depender daquilo que está sendo apurado, poderá fornecer algum auxílio no exame, para depois tentar realizar a extração do cartão de memória, caso ainda não tenha sido realizada (e.22, e.23 e e.24)

A Figura 4.6 ilustra os processos de extração dos dados de um dispositivo Android com controle de acesso ativado.

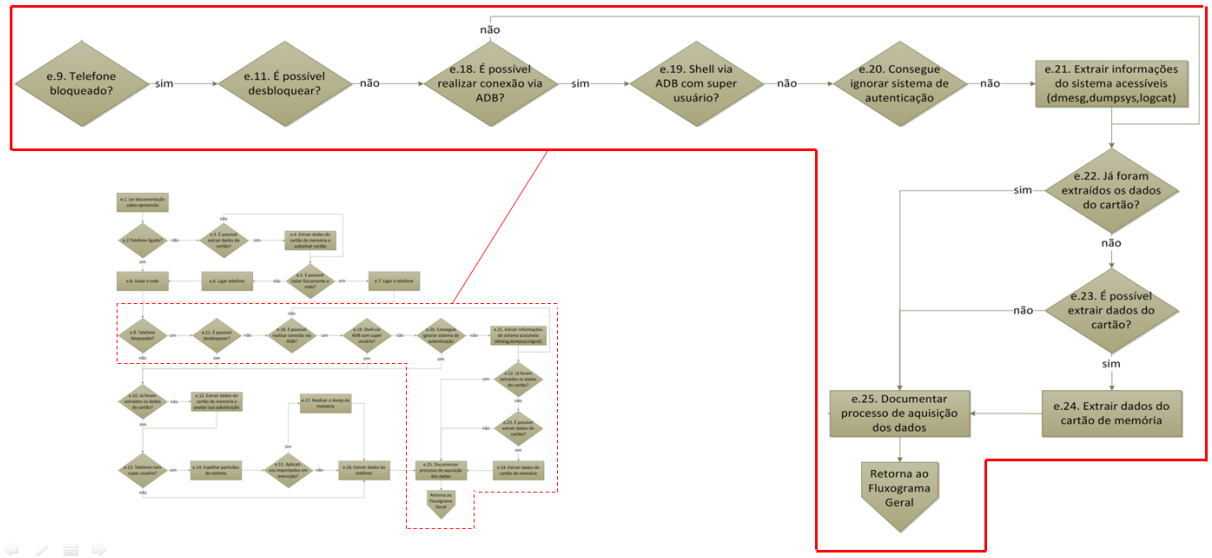


Figura 4.6 – Processos que envolvem um *smartphone* com controle de acesso ativado.

4.2.4. Considerações sobre a aquisição

Todo o processo de aquisição dos dados deve ser documentado (e.25) com suas particularidades e ressalvas, descrevendo também as informações sobre o sistema Android, como sua versão e *kernel*. A correta documentação de todos os procedimentos será essencial para o processo de exame (“f. Exame”) que será a etapa seguinte, conforme ilustrado no fluxograma da Figura 4.1, sendo também necessário estar descrito no relatório/laudo todos os procedimentos realizados com a finalidade de esclarecer às partes a forma como os dados foram adquiridos do equipamento apreendido.

4.3. EXAME

Os processos de um exame de um *smartphone* com o sistema Android encontram-se ilustrados no fluxograma da Figura 4.7. Antes de iniciar a análise daquilo que foi extraído na etapa anterior, o analista pericial deve se preocupar em definir os objetivos do exame (f.1), baseando-se no que foi solicitado no processo "b. Encaminhar a exame" do fluxograma da Figura 4.1. Esta definição é importante, pois, a depender do que se está sendo apurado, o exame pode seguir paradigmas diferentes na análise dos dados extraídos, podendo, por exemplo, o foco ser apenas imagens e vídeos ou contatos e geolocalização.

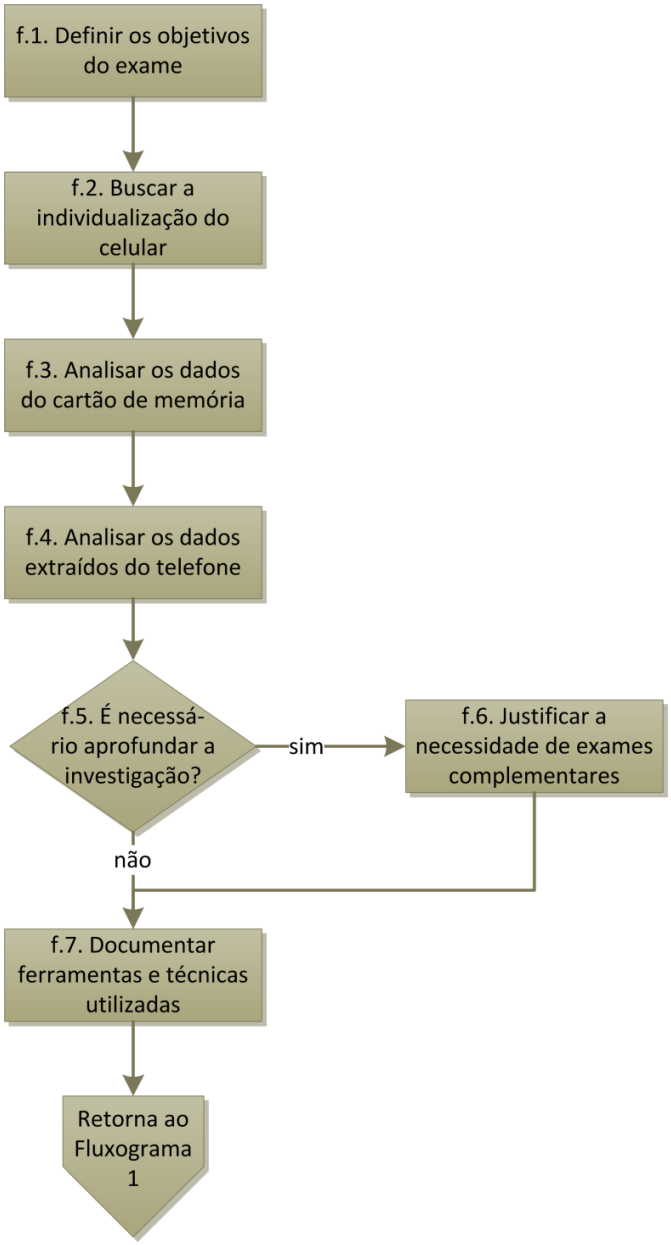


Figura 4.7 - O exame de um *smartphone* com a plataforma Android.

4.3.1. A individualização do *smartphone*

Definido os objetivos do exame, o especialista deve buscar nos dados extraídos, e até mesmo no próprio *smartphone* quando necessário, informações que possam definir o proprietário do aparelho, individualizando-o (f.2). São realizadas buscas nos dados extraídos, a exemplo do nome de usuário da conta Google usada, nome do endereço de e-mail, usuários de comunicadores instantâneos, anotações da agenda, cartões de visita digitais, dentre outras. Esta individualização do telefone determina quem é o usuário do aparelho, podendo assim vincular as evidências e provas encontradas na análise a um suspeito de forma inquestionável.

4.3.2. A análise dos dados do dispositivo

Como os telefones celulares com o sistema Android possuem cartões de memória, a análise deve iniciar pelos dados extraídos destas mídias (f.3). A partir de uma imagem do cartão de memória, que foi obtida na fase de aquisição (fluxograma da Figura 4.2), é possível utilizar as ferramentas forenses normalmente utilizadas para computadores para visualização da estrutura de arquivos, realizar buscas por palavras chave, buscas por expressões regulares, visualização das imagens e vídeos, ou seja, realizar o exame buscando atingir o objetivo definido.

A partir daí, o exame dos dados extraídos do *smartphone* (f.4) pode variar a depender da forma como foram obtidos. Se foram extraídos a partir de uma ferramenta forense, deve-se analisar a saída produzida, observando o relatório gerado e os arquivos gerados e recuperados. Normalmente, ferramentas específicas para uma determinada plataforma conseguem atingir bons resultados, uma vez que simula a extração manual, automatizando o processo. Entretanto, na fase de aquisição, o analista deve ter realizado uma comparação daquilo que foi extraído pelo aplicativo com as informações constante no telefone, complementando o relatório gerado pela ferramenta forense.

Se os dados extraídos foram obtidos a partir de uma imagem do sistema por meio de um acesso de super usuário, o examinador pode-se valer de editores hexadecimais e ferramentas forenses usadas para análise do cartão de memória, e outras técnicas periciais, a fim de

realizar a análise. Também pode valer de *disassemblers*[21] de arquivos "dex" para auditar aplicativos instalados.

A fim de realizar a análise do banco de dados extraídos do cartão de memória ou da memória interna do telefone, o analista deve utilizar softwares do SQLite, uma vez que a plataforma Android adotou este banco de dados relacional como padrão (vide seção 3.7.3). A análise dos arquivos referentes ao SQLite são muito importantes, uma vez que praticamente todos os dados armazenados pelos aplicativos estão neste gerenciador de banco de dados. Assim, a depender da situação, é possível, por exemplo, obter informações a respeito dos mapas armazenados em *cache* pelo Google *Maps Navigation* (Hoog, 2010).

4.3.3. Aprofundando a investigação

Além da grande capacidade de armazenamento de informação, os *smartphone*s com o sistema operacional Android instalado possuem muitas funcionalidades associadas à computação em nuvem. Dada esta característica, pode ser interessante à investigação que os exames sejam aprofundados, buscando informações que estão além da barreira física imposta pelo dispositivo.

Encontrando informações como fotos, vídeos, arquivos, anotações, contatos, e-mails, favoritos, dentre outras, que estão armazenadas em serviços oferecidos na Internet, o analista pericial responsável pelos exames deve avaliar a necessidade de informar a equipe de investigação a utilidade que tais dados podem ter ao apuratório (f.5). Ademais, pode haver a situação em que os dados do telefone não puderam ser extraídos, dada a inviabilidade de desbloqueá-lo (processos e.11 e e.18 à e.20 da Figura 4.3), podendo ser necessária a utilização de técnicas mais invasivas para acesso ao telefone. Assim, quando for uma situação que o examinador acredite ser essencial para complementação aos exames, este pode emitir um documento informando a necessidade dos exames complementares (f.6). Assim, a autoridade responsável pela investigação poderá avaliar, juntamente com o analista pericial, ciente das informações repassadas por esse, a necessidade de se providenciar os meios para que novos exames sejam realizados, a exemplo de uma ordem judicial para ter acesso a uma caixa de e-mails ou a um repositório de arquivos *online*, ou permissão para realizar procedimentos que possam danificar o aparelho.

---

[21] Alguns exemplos são: dex2jar (http://code.google.com/p/dex2jar), baksmali (http://code.google.com/p/smali), dedexer (http://dedexer.sourceforge.net); acessados em 22/05/2011.

4.3.4. Considerações sobre o exame

Assim como no processo de apreensão e de aquisição dos dados, todo o processo de exame deve ser documentado (f.7). Ao relatar o que foi realizado no exame, o especialista deve usar uma linguagem bem clara e objetiva, pois o que foi encontrado nesta etapa dará os subsídios para formação da conclusão do trabalho.

5. ESTUDO DE CASOS

A fim de validar o método proposto no capítulo 4, esta parte do trabalho é dedicada a apresentar estudo de casos a partir de três *smartphones* com o sistema operacional Android instalado.

Serão apresentados três cenários distintos. O primeiro e o segundo foram selecionados objetivando relatar a realidade mais comum quando um telefone celular é apreendido e tem seus dados extraídos e examinados. O terceiro cenário objetiva visualizar uma situação mais complexa que um analista pericial pode se deparar no momento de realizar a extração e o exame do celular, quando um telefone está bloqueado, com permissões de super usuário e acesso de depuração habilitados.

5.1. CONSIDERAÇÕES SOBRE OS CENÁRIOS

Os cenários partem da premissa que a solicitação dos exames objetiva apenas extrair os dados armazenados no dispositivo, e que não havia um analista pericial na equipe de apreensão. Esta, normalmente, é a realidade na casuística da Polícia Federal, onde são formadas equipes de busca e apreensão com Agentes e Escrivães sob a coordenação de um Delegado, sem a presença de Peritos. A Figura 5.1 ilustra os processos e decisões que uma equipe de busca se deparará no caso em que não há a presença de um especialista.

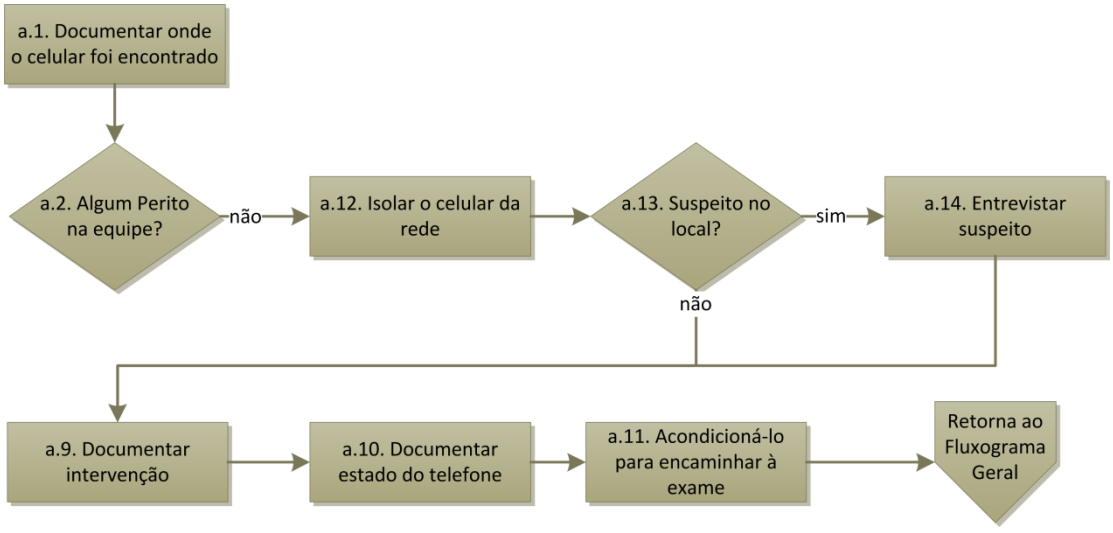


Figura 5.1 – Apreensão de um telefone celular sem a presença de um analista pericial.

Será considerado como objetivo dos exames, a extração das informações julgadas como relevantes pelo analista pericial, sejam elas mensagens, registros de chamadas, e-mails, imagens, vídeos, dentre outras mais que possam alimentar o procedimento investigativo dado o equipamento apreendido.

Nos cenários propostos não serão utilizados invólucros de isolamento de sinal, salas de isolamento de sinal ou gaiolas de Faraday, tendo em vista a ausência destes dispositivos na realização dos testes.

Serão utilizados três modelos distintos de *smartphones* em cada um dos cenários propostos. As características dos equipamentos encontram-se descritas na Tabela 5.1.

Tabela 5.1 - Descrição dos cenários propostos nos estudos de caso.

	Características do *smartphone*				
	Ligado	Bloqueado	Acesso de depuração	Super usuário	Descrição
Cenário 1	Sim	Não	Não	Não	Sony Ericsson, modelo Xperia X10 miniPro U20a, Android versão 2.1, de um usuário normal[22];
Cenário 2	Não	Sim	Não	Não se aplica	Motorola, modelo Milestone II A953, Android versão 2.2.1, de um usuário avançado[23]
Cenário 3	Sim	Sim	Sim, restrito	Sim	Motorola, modelo Milestone A853, Android versão 2.2.2, de um usuário expert[24]

5.2. CENÁRIO 1: APARELHO LIGADO, SEM BLOQUEIO E SEM PERMISSÕES DE SUPER USUÁRIO

Este cenário tem como objetivo apresentar uma situação de exame de um *smartphone* de um usuário normal que não se encontrava no local da apreensão e utiliza apenas as aplicações básicas, sem controle de acesso ativado e tampouco permissões de super usuário instaladas.

O *smartphone* foi apreendido por uma equipe policial conforme ilustrado na Figura 5.1. Será considerado que o suspeito não se encontrava no local no momento da apreensão e que o equipamento não foi desligado ao ter sido isolado da rede, sendo colocado em modo avião.

[22] Utiliza apenas as aplicações básicas do telefone celular.
[23] Utiliza outras aplicações além daquelas fornecidas por padrão pelo telefone celular, integrando-o ao seu cotidiano.
[24] Utiliza todos os recursos de hardware e software do aparelho de telefonia celular, com conhecimentos para instalação de customizações no sistema, podendo até mesmo a desenvolver aplicações específicas.

5.2.1. A aquisição dos dados

A Figura 5.2 ilustra o fluxo a ser seguido no processo de aquisição dos dados ante o cenário descrito, onde o telefone não possui bloqueio nem permissões de super usuário.

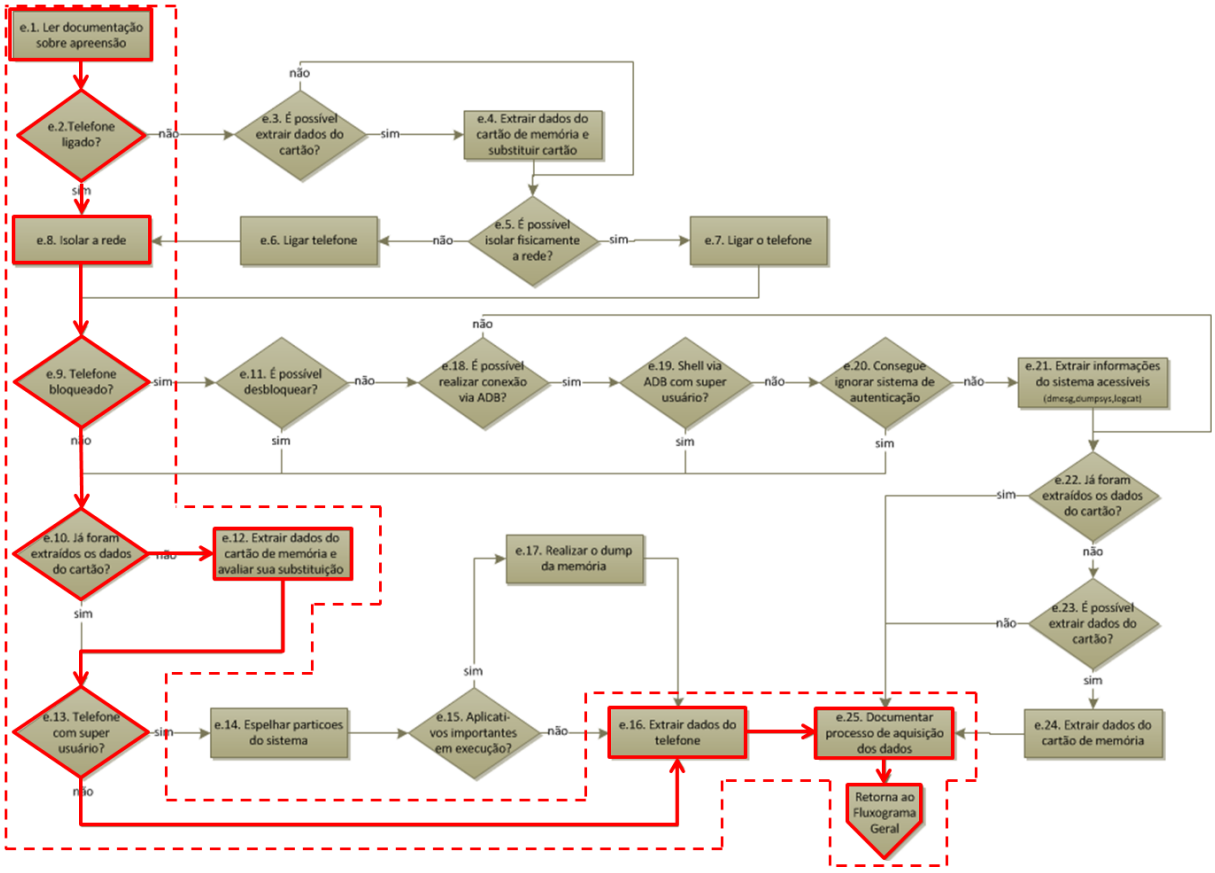


Figura 5.2 - Aquisição de dados de um telefone sem bloqueio e sem permissões de super usuário.

No processo e.1, o analista pericial obtém a informação de que o celular foi colocado em modo avião, que o suspeito não se encontrava no local, assim como obterá as descrições do dispositivo, aferindo que trata-se do mesmo equipamento que está sendo encaminhado para extração dos dados.

O analista constata que o telefone encontra-se ligado (e.2), não havendo necessidade de isolá-lo da rede (e.8), pois já se encontrava com modo avião ativado, conforme descrito na documentação do processo e.1.

No processo e.9, constata que o celular não se encontra bloqueado. A partir daí, como os dados do cartão de memória ainda não foram extraídos (e.10), este será o momento para realizar este procedimento. O modelo do *smartphone* deste cenário, Sony Xperia X10 mini

Pro, não permite a remoção física do cartão de memória. Isso justifica conectar o celular à estação de trabalho[25] e selecionar, no sistema, o gerenciamento do cartão de memória via USB. Foi utilizada a ferramenta da Access Data, FTK Imager, versão 3.0.1, para realizar a extração dos dados do cartão de memória (e.12). Abrindo a ferramenta FTK Imager, é possível obter uma cópia no formado "dd" do cartão de memória que foi denominada "cenario1.dd", ilustrado na Figura 5.3, gerando o respectivo código de integridade.

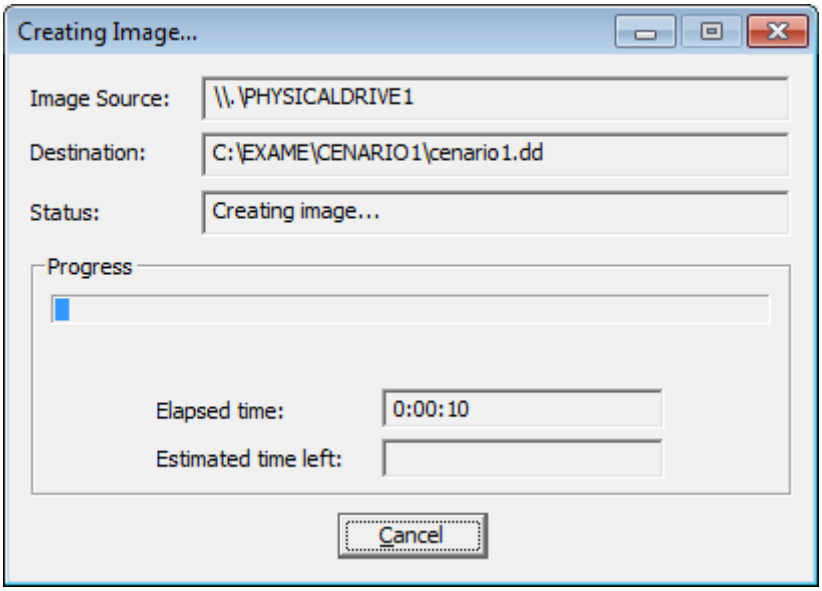


Figura 5.3 - Cópia dos dados contidos no cartão de memória para a estação de trabalho do perito.

Após a realização da cópia integral do cartão de memória original para o arquivo "cenario1.dd", não é possível o especialista substituí-lo, uma vez que não é removível, o que justifica o seu uso no decorrer da extração.

Navegando pelo sistema, listando os aplicativos instalados, nota que não há permissão de super usuário (e.13) e segue para extração dos dados do telefone (e.16). Foi utilizado o aplicativo da Via Forensics, "Android Forensics Logical Application" (Hoog, 2010), e inspeção visual de forma manual por meio da navegação pelo sistema.

Primeiramente deve ser instalada a aplicação no sistema Android. Para isso, foi ativado o modo de depuração USB, no menu de configurações de aplicativos, desenvolvimento, opção depuração USB (Figura 5.4 'a' e 'b').

[25] A porta USB deve estar configurada para proteger o dispositivo de escrita.

(a) (b)

Figura 5.4 - Ativação do modo de depuração USB para conexão via ADB.

Assim, é possível ao analista instalar aplicativos no telefone por meio da ferramenta ADB disponibilizada pelo SDK do Android, conforme Figura 5.5.

```
C:\android-sdk\platform-tools>adb devices
List of devices attached
42593930303935485637    device

C:\android-sdk\platform-tools>adb   -s   42593930303935485637   install
AndroidForensics.apk
329 KB/s (31558 bytes in 0.093s)
        pkg: /data/local/tmp/AndroidForensics.apk
Success
```

Figura 5.5 - Instalação do aplicativo da Via Forensics por meio da ferramenta ADB do SDK do Android.

Executando o “Android Forensics Logical Application” já instalado no telefone, este fornece um menu de opções daquilo que será extraído do sistema (Figura 5.6), gerando arquivos “.csv” (*comma-separated values*) no cartão de memória do examinador, em uma pasta denominada */forensics*.

Estes arquivos gerados pelo aplicativo forense são analisados, realizando uma comparação com aquilo que pode ser extraído manualmente do telefone, a fim de complementar qualquer informação adicional que por ventura possa ter. No cenário em questão foram encontradas várias anotações textuais no aplicativo “Observações” que não foram extraídas pelo aplicativo da Via Forensics. As anotações foram extraídas manualmente uma vez que possuíam informações a respeito de reserva em pousada, um telefone de uma pessoa que não se encontrava nos contatos e cupons de reserva em um restaurante da cidade. Também foi

observado que o formato da data armazenada pelo "Android Forensics Logical Application" segue um padrão com 13 dígitos[26], que foram comparados com as datas nos registros do telefone.

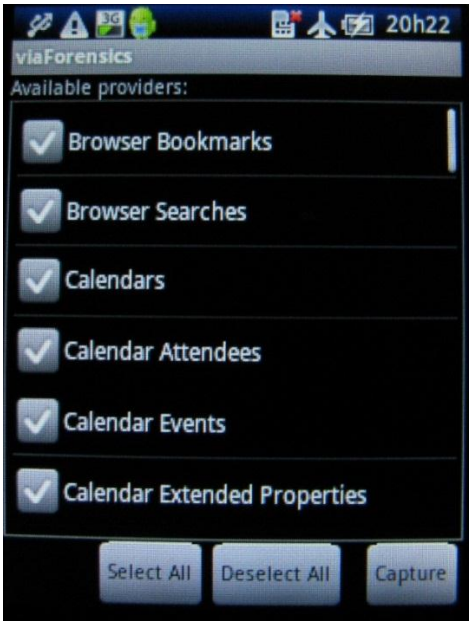


Figura 5.6 - Aplicativo da Via Forensics e as opções de extração dos dados.

A seguir todo o processo de extração é documentado (e.25), onde é esclarecido que no cenário em questão não foi necessário isolar o dispositivo da rede uma vez que este já se encontrava em modo avião; que a extração do cartão de memória ocorreu sem falhas gerando um código de integridade; que não foi possível substituir o cartão original por um cartão do examinador; que foi instalado um aplicativo forense no telefone apreendido para extração dos dados armazenados e; que uma inspeção visual foi realizada com a finalidade de complementar os dados extraídos. Os dados extraídos estão descritos na seção 5.2.2, que trata da realização dos exames.

No processo de documentação deve-se anotar as contas configuradas no dispositivo, assim como informar que se encontrava sem sincronização automática. Também são colhidas informações sobre o telefone a partir no menu de configuração, como: o modelo, U20a; versão do firmware, 2.1-update1; versão do *kernel*, 2.6.29SEMCUser@SEMCHost #1; número da versão, 2.0.A.0.504; IMEI, 012343001959006 e; o momento em que foi ligado, dia 27/04/2011 05:54 (horário de Brasília).

[26] Este formato é utilizado pelo sistema Android como um todo para armazenar as datas. No Microsoft Office Excel 2010 foi necessário formatar a célula para data, dividir o valor por 86400000 e somar 25569, para obter a data em UTC.

Assim é possível fornecer subsídios à fase de exame, bem como descrever os procedimentos realizados e seus resultados no relatório/laudo que será redigido ao final do processo de análise pericial.

5.2.2. Exame

Definido que o objetivo dos exames é a extração completa de todas as informações do *smartphone*, busca-se determinar o indivíduo que fazia o seu uso (processo f.2 da Figura 4.7).

Neste cenário, a individualização do telefone ocorreu no momento da extração, quando foi possível obter a conta configurada no dispositivo, crissmunhozinni71@gmail.com[27], que aparenta ser o nome do proprietário. Outra informação obtida encontra-se armazenada nos contatos do telefone, onde há os registros "Casa", "Casa 2", "Mamãe" e "Papai", que podem facilmente ser associados a proprietários de linhas de telefones.

Foi possível obter imagens dos dados examinados do cartão de memória (processo f.3 da Figura 4.7), dentre elas, fotos tiradas que aparentam ser da família. Não foi possível obter dos metadados das fotos as coordenadas GPS de onde elas foram tiradas. Também foram encontrados arquivos de música do tipo mp3.

A partir dos dados extraídos do *smartphone* (processo f.4 da Figura 4.7), foi possível obter os contatos da agenda telefônica. Foi observado que o usuário fazia pouco uso das mensagens SMS e usava o calendário com registros de compromissos com frequência, podendo inclusive realizar um levantamento de quando supostamente esteve na academia, na farmácia e no teatro. Além disso, foram obtidos 500 registros de ligações recebidas, realizadas e não atendidas, assim como o conteúdo dos arquivos de texto obtidos a partir do aplicativo "Observações" no processo e.16 da Figura 5.2.

Uma vez que se trata de um *smartphone* de um usuário normal sem grandes conhecimentos sobre a plataforma, não há informações disponíveis que possam justificar exames adicionais (processo f.5 da Figura 4.7). Assim é realizada a documentação das ferramentas e técnicas utilizadas no exame. A Figura 5.7 ilustra a etapa de realização dos exames.

[27] O e-mail foi alterado a fim de preservar o endereço verdadeiro.

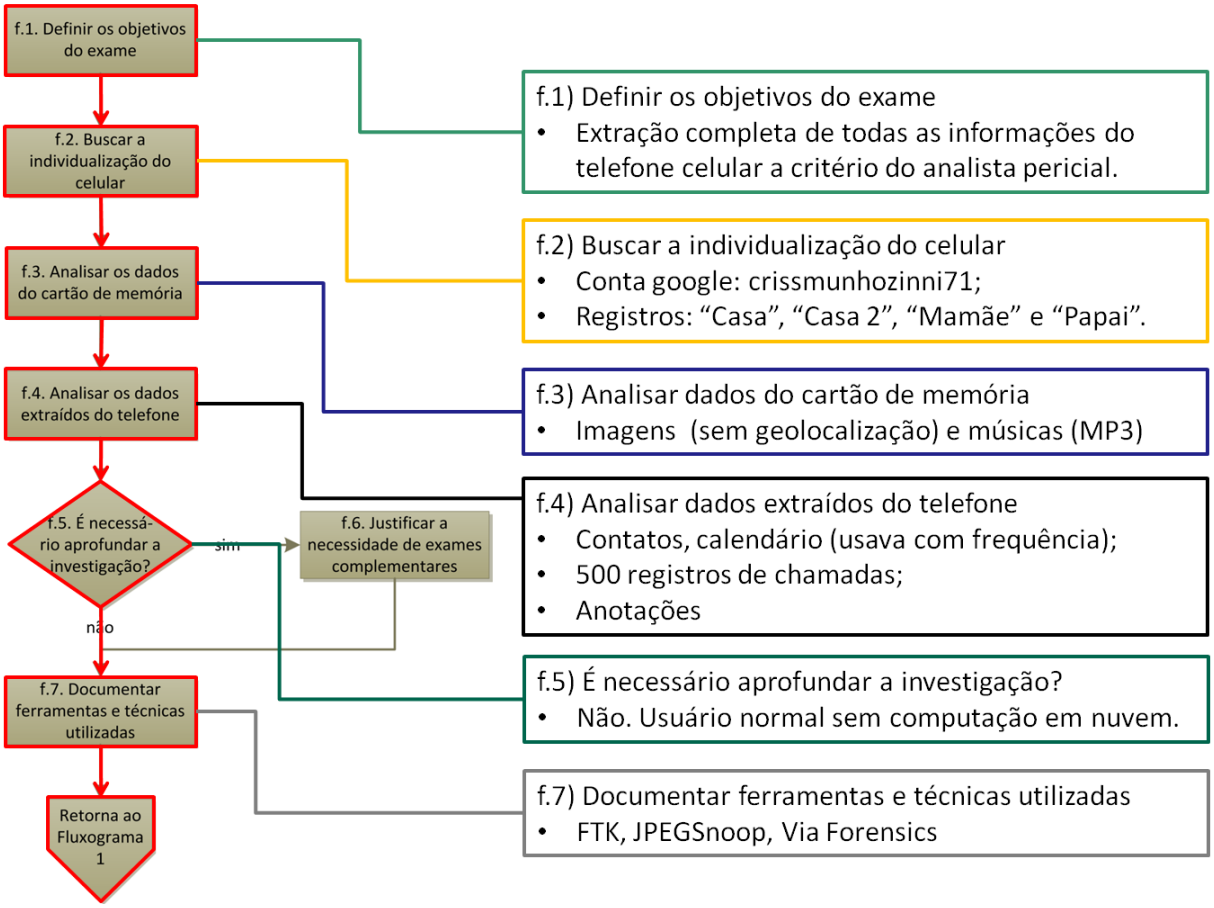


Figura 5.7 – Etapa de exame do cenário 1 (Sony Ericsson Xperia X10 miniPro).

Para realização do exame, foi utilizada a ferramenta Forensic Toolkit da Access Data, versão 1.81, que possui um visualizador hexadecimal e sistema de busca indexada por palavras-chave. No cartão de memória foi possível buscar os dados apagados que puderam ser recuperados por meio da técnica de *carving*. Entretanto, não foram encontrados dados apagados relevantes para investigação. Os metadados das fotos obtidas do cartão de memória foram analisados com a ferramenta JPEGSnoop, versão 1.5. Entretanto, não foi possível obter as coordenadas GPS de onde foram tiradas. Também foram realizadas análise diretamente no *smartphone* buscando por informações que pudessem estar armazenadas em aplicativos específicos a exemplo do aplicativo "Observações", assim como foram analisados os relatórios gerados pelo aplicativo "Android Forensics Logical Application", que extraiu informações importantes do telefone apreendido.

5.3. CENÁRIO 2: APARELHO DESLIGADO, COM BLOQUEIO E SEM ACESSO DE DEPURAÇÃO USB (ADB)

Este cenário tem como objetivo apresentar uma situação de exame de um *smartphone* de um usuário avançado, que não se encontrava no local da apreensão, e utiliza outras aplicações além daquelas fornecidas por padrão pelo *smartphone*, integrando-o ao seu cotidiano, com controle de acesso ativado e sem acesso de depuração USB ativado (ADB).

O telefone celular foi apreendido por uma equipe policial conforme ilustrado na Figura 5.1. Será considerado que o suspeito não se encontrava no local no momento da apreensão e que o equipamento foi desligado para ser sido isolado da rede.

5.3.1. A aquisição dos dados

A Figura 5.8 ilustra o fluxo a ser seguido no processo de aquisição dos dados, dado o cenário descrito, em que o telefone possui bloqueio e não tem permissão de super usuário habilitada.

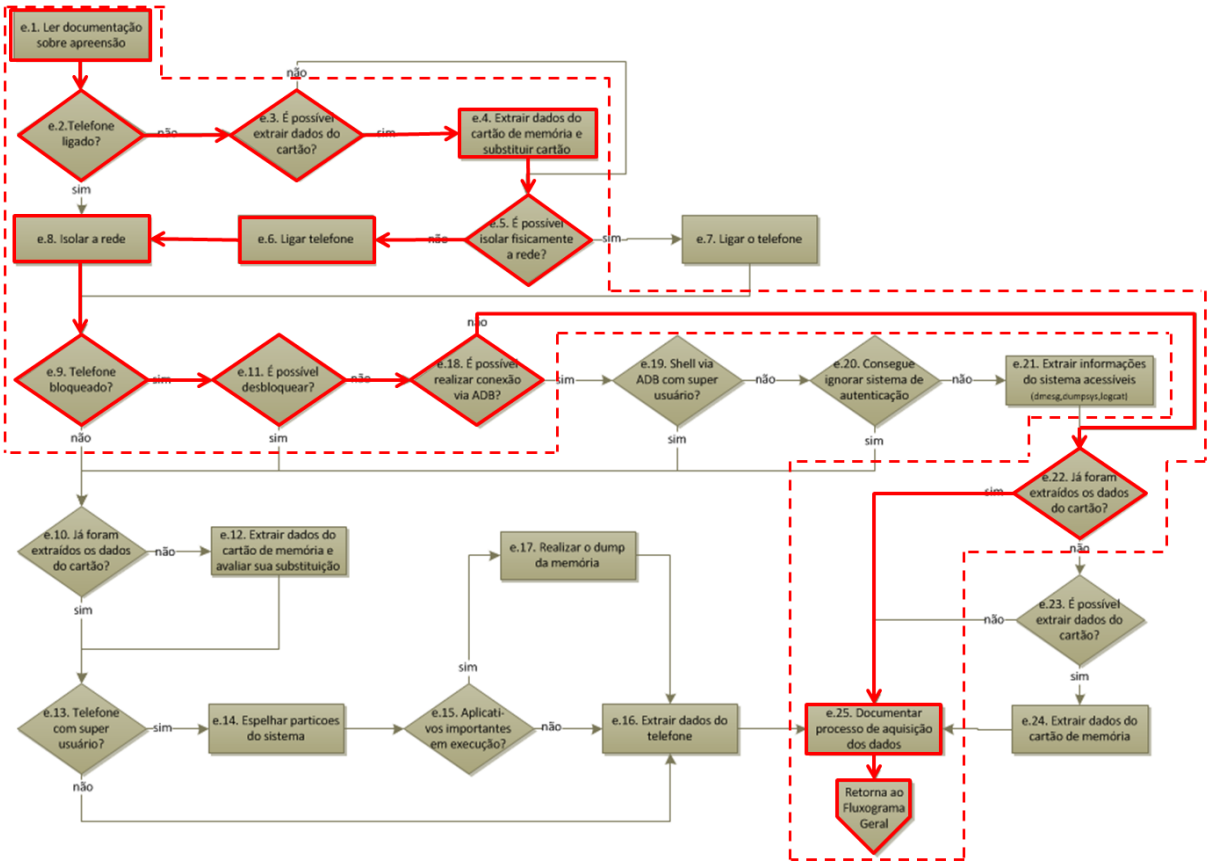


Figura 5.8 - Fluxo do processo de aquisição de dados de um *smartphone* encaminhado desligado, com bloqueio e sem acesso ADB.

No processo e.1, o analista pericial obtém a informação de que o celular foi desligado no momento da apreensão, que o suspeito não se encontrava no local, assim como obterá as descrições do dispositivo, aferindo que trata-se do mesmo equipamento que está sendo encaminhado para extração dos dados.

O analista constata que o telefone encontra-se desligado (e.2) e verifica que neste modelo, Motorola Milestone II, é possível remover o cartão de memória para extração dos dados antes de ter que ligar o aparelho (e.3).

Assim o cartão é removido do dispositivo, inserido no leitor de cartão e conectado à porta USB da estação pericial, com proteção de gravação. Abrindo a ferramenta FTK Imager, é possível obter uma cópia no formado "dd" do cartão de memória que foi denominada "cenario2.dd", similar ao ilustrado na Figura 5.3, gerando o respectivo código de integridade (e.4).

Após a realização da cópia integral do cartão de memória original para o arquivo "cenario2.dd", o especialista deve clonar este conteúdo em um cartão de memória do examinador e verificar se há espaço suficiente no cartão de memória do examinador para extração dos dados do telefone, conforme Figura 5.9. Neste caso, foi utilizado o comando "dd", espelhando todo conteúdo do cartão original para o cartão do examinador, a partir da ferramenta "Cygwin" para ambiente Windows.

```
$ dd if=cenario2.dd.001 of=\\.\PHYSICALDRIVE1
15548416+0 records in
15548416+0 records out
7960788992 bytes (8.0 GB) copied, 1425.82 s, 5.6 MB/s
```

Figura 5.9 - Cópia do conteúdo do cartão original para o cartão do examinador.

Uma vez que o analista pericial não dispõe de um laboratório com isolamento de sinal (e.5), é necessário ligar o *smartphone* (e.6) e colocá-lo imediatamente em modo avião a fim de isolá-lo da rede de telefonia (e.8), documentando qualquer alteração que porventura ocorra neste procedimento (por exemplo o recebimento de uma mensagem SMS), fato que não ocorreu no cenário em questão.

Com o celular ligado, observa-se que este se encontra bloqueado (e.9), e não é possível desbloqueá-lo (e.11). Desta forma, tentou-se realizar uma conexão da estação pericial com o telefone apreendido via ADB (Figura 5.10), entretanto sem sucesso (e.18).

```
C:\android-sdk\platform-tools>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached

C:\android-sdk\platform-tools>
```

Figura 5.10 - Tentativa de acesso via ADB ao celular bloqueado.

Dado este cenário em que o telefone encontra-se bloqueado sem possibilidade de acesso via ADB, com o cartão de memória já clonado, resta ao analista documentar o processo de aquisição para dar prosseguimento ao exame. As informações extraídas estão descritas na seção 5.3.2, que trata da realização dos exames.

Logo, todo o processo de extração é documentado, quando é esclarecido que no cenário em questão, o analista recebeu o telefone desligado, sendo providenciada a extração dos dados do cartão de memória. Em seguida, o telefone foi ligado e colocado em modo avião para isolar o dispositivo da rede de telefonia. Posteriormente, observou-se que ele encontrava-se bloqueado e sem acesso via ADB, não sendo viável realizar a extração dos dados do telefone. Desta forma o analista fornece subsídios à fase de exame, assim como descreve os procedimentos realizados e seus resultados no relatório/laudo que será redigido ao final do processo de análise pericial.

5.3.2. Exame

Após definido que o objetivo dos exames é a extração completa do telefone celular de todas as informações úteis ao apuratório, a critério do analista pericial, busca-se individualizar o *smartphone* (processo f.2 da Figura 4.7).

Neste cenário foi possível observar que havia, na pasta *bluetooth* do cartão de memória, 60 arquivos do tipo ".vcf", denominados cartões de visita. Provavelmente haviam sido enviados ao telefone apreendido via *bluetooth* e incorporados aos contatos do telefone. Foi possível observar o arquivo "Casa.vcf" com um telefone fixo armazenado, o que indica a possível residência do proprietário do *smartphone* (processo f.2 da Figura 4.7).

Partindo para uma análise mais aprofundada dos dados do cartão de memória (processo f.3 da Figura 4.7), foram encontradas duas fotografias, tiradas em 20/04/2011 às 19:16 (horário de Brasília), que possuíam em seus metadados as coordenadas geográficas: latitude 15º 48' 0.000" sul e longitude 47º 53'0.000" oeste, que apontam para uma região localizada na cidade de Brasília, próximo ao Setor de Autarquias Sul no início do bairro da Asa Sul.

Também foram encontrados arquivos que dizem respeito ao aplicativo “Busybox”, o que indica uma grande possibilidade do celular estar com permissões de super usuário instaladas, uma vez que tal aplicativo é utilizado somente em celulares com estas permissões configuradas. Essa suspeita foi reforçada ao ser encontrado na raiz do cartão de memória o aplicativo “z4root.1.3.0.apk”, que ao ser instalado habilita permissões de super usuário no sistema.

Observou-se também que no cartão de memória havia os arquivos de *cache* do “Picasa”, o que significa que o usuário do telefone utiliza os serviços da Google para armazenamento de imagens da Internet (http://picasaweb.google.com).

Com relação aos dados armazenados no *smartphone* (processo f.4 da Figura 4.7), não foi possível analisá-los, uma vez que o sistema de controle de acesso do Android encontrava-se ativado e o acesso via ADB não se encontrava habilitado, inviabilizando a extração dos dados.

Uma vez que se trata de um *smartphone* bloqueado, em que não foi possível acessar os dados contidos no telefone, pode ser necessária a utilização de técnicas mais invasivas de acesso à memória interna do equipamento. Este procedimento deve ser discutido e avaliado com a autoridade solicitante dos exames uma vez que, a depender da marca e modelo, pode danificar os dados armazenados no *smartphone* (processos f.5 e f.6 da Figura 4.7). A Figura 5.11 ilustra a etapa de exame do cenário descrito.

A fim de documentar as ferramentas e técnicas utilizadas (processo f.7 da Figura 4.7), é necessário esclarecer que, para a realização do exame, foi utilizada a ferramenta Forensic Toolkit da Access Data (AccessData, 2011), versão 1.81, que possui um visualizador hexadecimal e sistema de busca indexada por palavras-chave. No cartão de memória foi possível buscar os dados apagados que puderam ser recuperados por meio da técnica de *carving*. Entretanto, não foram encontrados dados apagados relevantes para investigação. Foram encontradas fotografias com coordenadas geográficas nos metadados, e concluiu-se que o telefone provavelmente encontra-se com permissões de super usuário habilitadas. Não foram realizadas análise diretamente no *smartphone* uma vez que o acesso encontrava-se bloqueado. Deve-se, esclarecer quais informações são necessárias para o desbloqueio e se há possibilidade de desbloqueá-lo usando técnicas mais intrusivas.

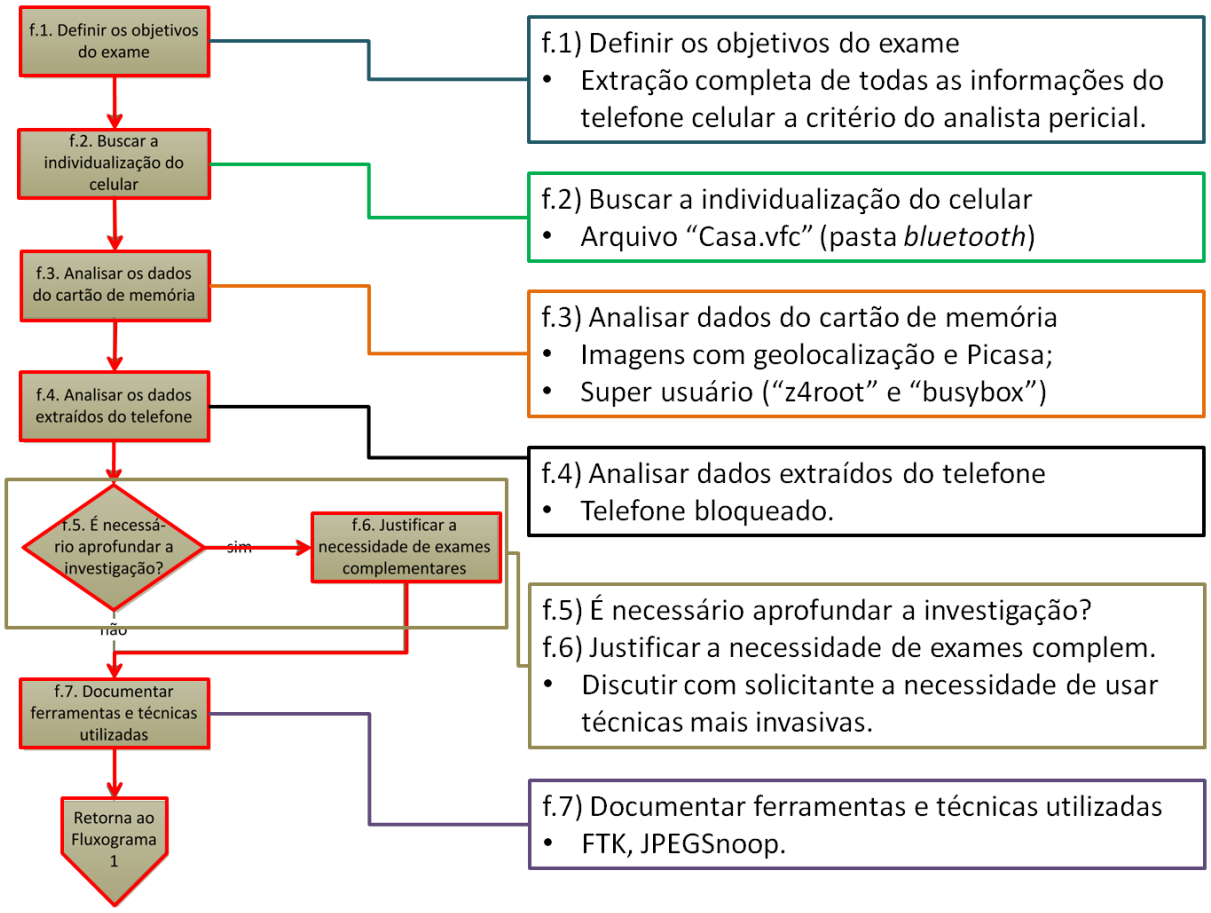


Figura 5.11 – Etapa de exame do cenário 2 (Motorola Milestone II).

5.4. CENÁRIO 3: APARELHO LIGADO, COM TELA DE BLOQUEIO, COM ACESSO DE DEPURAÇÃO RESTRITO (ADB), E COM PERMISSÕES DE SUPER USUÁRIO

Este cenário tem como objetivo apresentar uma situação de exame de um *smartphone* de um usuário expert, que não se encontrava no local da apreensão, e utilizava bem os recursos de hardware e software do *smartphone*, com conhecimentos para instalação de customizações no sistema, com controle de acesso ativado, acesso de depuração (ADB) ativado, mas restrito, e com permissões de super usuário instaladas.

O telefone celular foi apreendido por uma equipe policial, conforme ilustrado na Figura 5.1. Será considerado que o suspeito não se encontrava no local no momento da apreensão e que o equipamento não foi desligado ao ter sido isolado da rede, sendo colocado em modo avião. Não será considerado o uso de invólucro de isolamento de sinal, uma vez que não é comum as equipes de apreensão ter este tipo de material disponível, principalmente quando não há um especialista presente.

5.4.1. A aquisição dos dados

A Figura 5.12 ilustra o fluxo a ser seguido no processo de aquisição dos dados, dado o cenário descrito, em que o telefone possui bloqueio, mas com acesso via ADB com usuário do sistema "shell" e com permissões de super usuário instaladas.

No processo e.1, o analista pericial obtém a informação de que o celular foi colocado em modo avião, que o suspeito não se encontrava no local, assim como obterá as descrições do dispositivo, aferindo que trata-se do mesmo equipamento que está sendo encaminhado para extração dos dados.

O analista constata que o telefone encontra-se ligado (e.2), não havendo necessidade de isolá-lo da rede (e.8), pois já se encontrava com modo avião ativado conforme descrito na documentação do processo e.1.

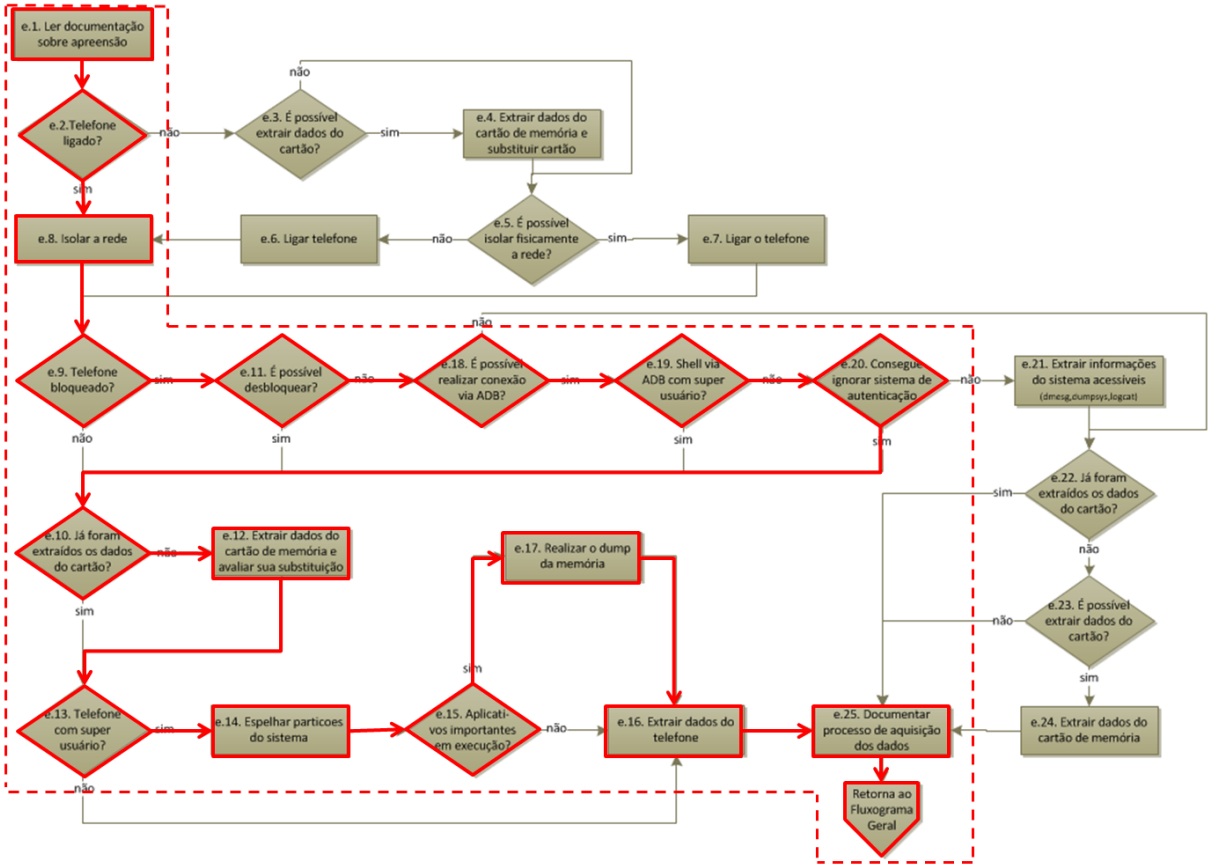


Figura 5.12 - Fluxo do processo de aquisição de dados de um *smartphone* Android ligado, bloqueado, com acesso ADB e permissões de super usuário.

Com o celular ligado, observa-se que este se encontra bloqueado (e.9), não sendo possível desbloqueá-lo (e.11). Neste cenário, o analista consegue, com sucesso, realizar uma conexão de depuração USB, da estação pericial com o telefone apreendido, via ADB (e.18).

Entretanto, isso se dá sem acesso ao *shell* com permissões de super usuário (e.19), conforme ilustrado na Figura 5.13.

```
C:\android-sdk\platform-tools>adb devices
List of devices attached
040140611301E014        device

C:\android-sdk\platform-tools>adb -s 040140611301E014 shell
$ su -
su -
Permission denied
$
```

Figura 5.13 - Obtenção de um shell via ADB no celular apreendido. Nota-se que não foi possível obter permissões de super usuário.

Com o acesso ADB restrito, é instalado o aplicativo "Screen Lock Bypass" (Cannon, 2011), e, em seguida, o aplicativo "Android Forensics Logical Application", conforme Figura 5.14, com a finalidade de ignorar o sistema de autenticação por padrão táctil do Android (e.20).

```
C:\android-sdk\platform-tools>adb -s 040140611301E014 install screenlockbypass.apk
224 KB/s (22797 bytes in 0.099s)
        pkg: /data/local/tmp/screenlockbypass.apk
Success

C:\android-sdk\platform-tools>adb -s 040140611301E014 install AndroidForensics.apk
716 KB/s (31558 bytes in 0.043s)
        pkg: /data/local/tmp/AndroidForensics.apk
Success
```

Figura 5.14 - Instalação dos aplicativos "Screen Lock Bypass" e "Android Forensics Logical Application" via ADB.

Como ainda não foram extraídos os dados do cartão, dado que o processo de remoção do mesmo poderia provocar perda de informações que ainda não tinham sido extraídas, nesse momento, serão copiados integralmente os dados do cartão (e.10).

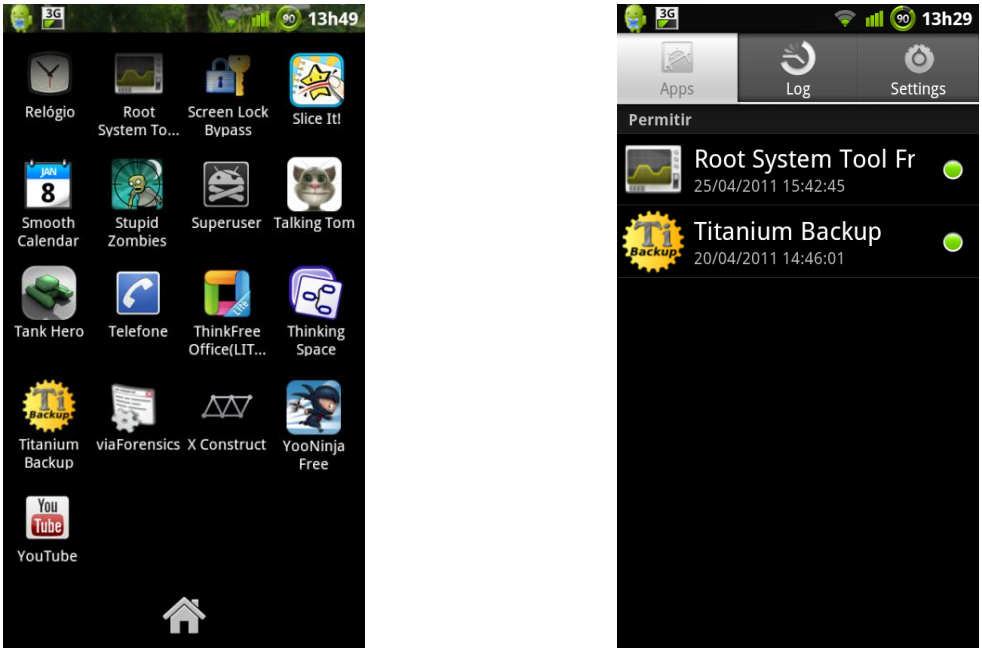
O modelo do *smartphone* deste cenário, Motorola Milestone A853, não permite a remoção física do cartão de memória sem a remoção da bateria, o que acarretaria o desligamento do aparelho. Para evitar o desligamento desnecessário, que não é desejável dado que se trata de um telefone de um usuário expert, o celular foi conectado à estação de trabalho[28] e foi selecionado no sistema o gerenciamento do cartão de memória via USB (Figura 5.15). Assim, foi utilizada a ferramenta da Access Data, FTK Imager, versão 3.0.1, para realizar a extração dos dados do cartão de memória conectado diretamente no aparelho (e.12). Abrindo a ferramenta FTK Imager, é possível obter uma cópia no formado "dd" do cartão de memória, que foi denominada "cenario3.dd", gerando o respectivo código de integridade.

[28] A porta USB deve estar protegida contra escrita.

Figura 5.15 - Tela de configuração da conexão USB.

Como a remoção do cartão acarreta um desligamento não desejável do *smartphone*, a extração dos dados do telefone é realizada diretamente no cartão original.

Navegando pelo sistema, observa-se que há instalado no *smartphone* o aplicativo "Superuser", que tem como finalidade fornecer permissões de super usuário (e.13), conforme ilustrado na Figura 5.16.



(a) (b)

Figura 5.16 - Aplicativo "Superuser": (a) programa no menu de aplicativos e (b) as permissões configurados no telefone.

Desta forma, é possível realizar uma nova conexão via ADB e conseguir um *shell* de super usuário para realizar o espelhamento das partições do sistema por meio do comando *dd* para o cartão de memória do examinador inserido no telefone (e.14), conforme Figura 5.17. Neste cenário são realizadas as cópias das partições *system*, *cache* e *userdata*. Entretanto, não há impedimentos para realização da cópia de todas as partições.

```
C:\Program Files\Android\android-sdk\platform-tools>adb -s 0401406113O1E014 shell
$ su -
# mount | grep mtd
/dev/block/mtdblock6 /system yaffs2 ro,relatime 0 0
/dev/block/mtdblock8 /data yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock7 /cache yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock5 /cdrom yaffs2 rw,relatime 0 0
/dev/block/mtdblock0 /pds yaffs2 rw,nosuid,nodev,relatime 0 0# cat /proc/mtd
cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00180000 00020000 "pds"
mtd1: 00060000 00020000 "cid"
mtd2: 00060000 00020000 "misc"
mtd3: 00380000 00020000 "boot"
mtd4: 00480000 00020000 "recovery"
mtd5: 008c0000 00020000 "cdrom"
mtd6: 0afa0000 00020000 "system"
mtd7: 06a00000 00020000 "cache"
mtd8: 0c520000 00020000 "userdata"
mtd9: 00180000 00020000 "cust"
mtd10: 00200000 00020000 "kpanic"
# ls /dev/mtd/mtd*
…
/dev/mtd/mtd6
/dev/mtd/mtd6ro
/dev/mtd/mtd7
/dev/mtd/mtd7ro
/dev/mtd/mtd8
/dev/mtd/mtd8ro
…
# df
/dev: 115788K total, 0K used, 115788K available (block size 4096)
/mnt/asec: 115788K total, 0K used, 115788K available (block size 4096)
/system: 179840K total, 118828K used, 61012K available (block size 4096)
/data: 201856K total, 172632K used, 29224K available (block size 4096)
/cache: 108544K total, 4908K used, 103636K available (block size 4096)
/dev: 115788K total, 0K used, 115788K available (block size 4096)
/mnt/asec: 115788K total, 0K used, 115788K available (block size 4096)
/cdrom: 8960K total, 8632K used, 328K available (block size 4096)
/tmp: 2048K total, 28K used, 2020K available (block size 4096)
/pds: 1536K total, 1356K used, 180K available (block size 4096)
/mnt/sdcard: 7770276K total, 5196760K used, 2573516K available (block size 4096)
# dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/_PERICIA/mtd6ro_system.dd bs=4096
dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/_PERICIA/mtd6ro_system.dd bs=4096
44960+0 records in
44960+0 records out
184156160 bytes transferred in 73.803 secs (2495239 bytes/sec)
# dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/_PERICIA/mtd7ro_cache.dd bs=4096
dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/_PERICIA/mtd7ro_cache.dd bs=4096
27136+0 records in
27136+0 records out
111149056 bytes transferred in 41.924 secs (2651203 bytes/sec)
# dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/_PERICIA/mtd8ro_userdata.dd bs=4096
dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/_PERICIA/mtd8ro_userdata.dd bs=4096
50464+0 records in
50464+0 records out
206700544 bytes transferred in 74.452 secs (2776292 bytes/sec)
# ls /mnt/sdcard/_PERICIA
ls /mnt/sdcard/_PERICIA
mtd6ro_system.dd
mtd7ro_cache.dd
mtd8ro_userdata.dd
```

Figura 5.17 - Copia das partições do sistema Android para o cartão de memória.

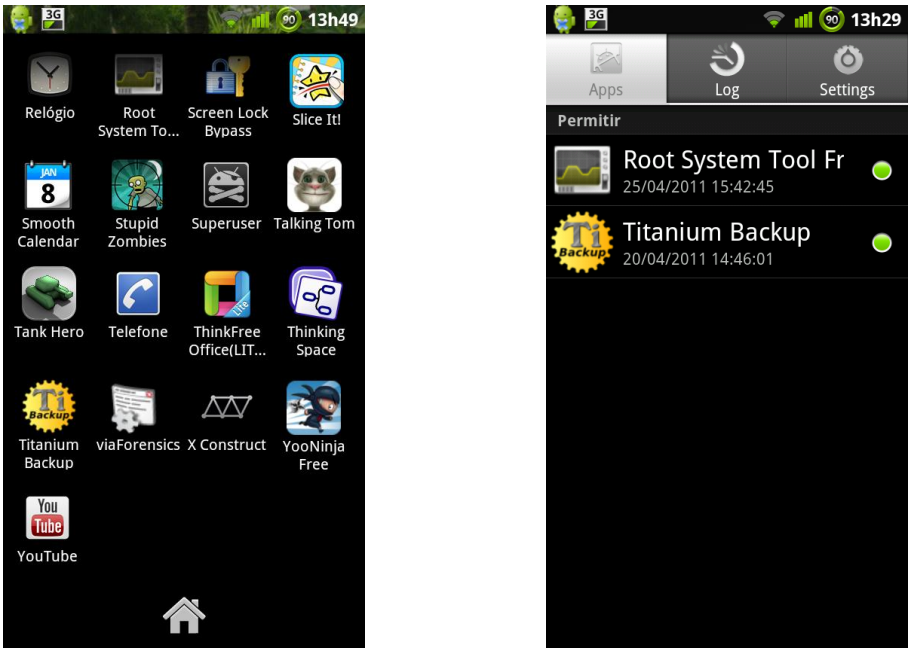
É importante o examinador usar o comando *mount* para anotar o tipo de sistema de arquivos usado nestas partições espelhadas a fim de facilitar sua análise no exame, que neste caso é o YAFFS2.

Nesta mesma *shell*, pode-se listar os processos dos aplicativos em execução no sistema (e.15), conforme Figura 5.18[29], observando que há aplicativos que podem fornecer informações importantes quando em execução.

```
C:\android-sdk\platform-tools>adb -s 040140611301E014 shell
$ su -
su -
# id
Id
uid=0(root) gid=0(root)
# ps | grep app
ps | grep app
USER     PID   PPID  VSIZE  RSS     WCHAN    PC         NAME
app_23            1703      1380      144116   11364   ffffffff   afd0ece8   S
com.android.inputmethod.latin
app_45    1712  1380  137784 12428 ffffffff afd0ece8 S com.motorola.usb
app_25    1717  1380  164352 24228 ffffffff afd0ece8 S com.android.launcher
app_18    1763  1380  171460 15656 ffffffff afd0ece8 S com.google.process.gapps
app_38            5767      1380      138112   10488   ffffffff   afd0ece8   S
smupdaterapp.service.UpdateCheckService
app_86            5797      1380      135828   10816   ffffffff   afd0ece8   S
se.catharsis.android.calendar
app_29            6199      1380      159552   12980   ffffffff   afd0ece8   S
com.google.android.apps.maps:NetworkLocationService
app_16            6221      1380      142064   10936   ffffffff   afd0ece8   S
com.google.android.apps.genie.geniewidget
app_68    6233  1380  142344 10464 ffffffff afd0ece8 S com.metago.astro
app_50    6290  1380  137964 10412 ffffffff afd0ece8 S nitro.phonestats
app_77            6309      1380      137940   10920   ffffffff   afd0ece8   S
net.rgruet.android.g3watchdog
app_73    6379  1380  144444 14796 ffffffff afd0ece8 S com.dropbox.android
app_13    6410  1380  136776 11448 ffffffff afd0ece8 S android.process.media
app_31            6429      1380      139204   11116   ffffffff   afd0ece8   S
com.google.android.apps.uploader
app_17    6440  1380  143804 12524 ffffffff afd0ece8 S com.google.android.gm
app_1     6445  1380  138592 13616 ffffffff afd0ece8 S android.process.acore
app_46    6453  1380  138096 12124 ffffffff afd0ece8 S com.android.vending
app_42    6504  1380  135668 15464 ffffffff afd0ece8 S com.noshufou.android.su
```

Figura 5.18 - Obtenção de um shell no telefone apreendido com acesso a super usuário e processos em execução.

[29] Foi utilizado o comando "ps | grep app" para limitar a saída dos processos em execução no sistema.



(a) (b)

Figura 5.19 - Aplicativo "Superuser": (a) programa no menu de aplicativos e (b) as permissões configuradas no telefone.

A partir do exposto na Figura 5.20, é possível obter os dados em memória dos aplicativos em execução no sistema (e.17). Como a finalidade do cenário é validar o método proposto, serão obtidos os dados da memória somente dos aplicativos Gmail (app_17, PID 6440), Dropbox (app_73, PID 6379), Mapas (app_29 PID 6199) e Calendário (app_86, PID 5797), usando a técnica apresentada por Thomas Cannon (Cannon, 2010), conforme Figura 5.18. Os arquivos de *dump* dos respectivos aplicativos são copiados para a estação pericial para posterior exame.

Para extração dos dados do telefone (e.16), foram copiados os conteúdos dos diretórios */data/system*, */data/misc* e */data/data* para o cartão de memória (vide seção 3.7.2), com a finalidade de facilitar a extração das configurações do sistema, configurações de hardware e informações dos aplicativos e seus bancos de dados, assim como os arquivos de imagem das partições do sistema que foram gravados no cartão de memória (vide Figura 5.17).

Depois se utilizou o aplicativo da Via Forensics, "Android Forensics Logical Application" (Hoog, 2010), instalado anteriormente no sistema; e inspeção visual de forma manual por meio da navegação pelo sistema.

Executando o "Android Forensics Logical Application", este fornece um menu de opções daquilo que será extraído do sistema, similar ao ilustrado na Figura 5.6, gerando arquivos ".csv" (*comma-separated values*) no cartão de memória do examinador, em uma pasta denominada "*forensics*".

```
# chmod 777 /data/misc
chmod 777 /data/misc
# kill -10 6440
kill -10 6440
# kill -10 6379
kill -10 6379
# kill -10 6199
kill -10 6199
# kill -10 5797
kill -10 5797
# ls /data/misc | grep dump
ls /data/misc | grep dump
heap-dump-tm1303909649-pid5797.hprof
heap-dump-tm1303909632-pid6199.hprof
heap-dump-tm1303909626-pid6379.hprof
heap-dump-tm1303909585-pid6440.hprof
# exit
exit
$ exit
exit

C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /data/misc/heap-
dump-tm1303909649-pid5797.hprof
2206 KB/s (2773648 bytes in 1.227s)
C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /data/misc/heap-
dump-tm1303909632-pid6199.hprof
2236 KB/s (3548142 bytes in 1.549s)
C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /data/misc/heap-
dump-tm1303909626-pid6379.hprof
1973 KB/s (3596506 bytes in 1.779s)
C:\android-sdk\platform-tools>adb -s 0401406113O1E014 pull /data/misc/heap-
dump-tm1303909585-pid6440.hprof
1968 KB/s (2892848 bytes in 1.435s)
```

Figura 5.20 - Mostra o comando para listar os processos, terminá-los de forma abrupta para geração do arquivo de dump e cópia destes arquivos para a estação pericial.

A fim de complementar qualquer informação adicional que por ventura possa haver, esses arquivos gerados pelo aplicativo forense são analisados, realizando uma comparação com aquilo que pode ser extraído manualmente do telefone. Neste cenário, foi observado que o proprietário do celular usava com frequência seu calendário, sendo que a ferramenta da Via Forensics não extraiu essas informações. Por meio de inspeção manual, foi possível extrair as informações relevantes do calendário do *smartphone*. Alternativamente, como o telefone possui acesso de super usuário, o banco de dados do aplicativo poderia ser obtido a partir da imagem gerada do sistema. Outra informação interessante que não foi extraída do sistema se refere à aplicação "3G Watchdog", que possui o histórico de uso dos dados 3G do telefone, oferecendo inclusive a opção de exportar os dados para um arquivo ".csv", que foi gerado e copiado para a estação pericial. Também foram obtidos os dados sobre a localização dos seus amigos cadastrados no aplicativo "Latitude", obtendo a localização do proprietário do telefone, André Morum, que se encontrava em Brasília/DF, SQS 303, no dia 26/04/2011, e dos seus amigos Juliana Simão e Marcos Munhoz, que também se encontravam em

Brasília/DF no dia 26/04/2011, só que este no aeroporto e aquela na SMHN Quadra 1. Os dados extraídos estão descritos na seção 5.4.2, que trata da realização dos exames.

A seguir, documentou-se todo o processo de extração (e.25). Relatou-se que, no cenário em questão, não foi necessário isolar o dispositivo da rede, pois já se encontrava em modo avião; que o aparelho se encontrava bloqueado, com acesso de depuração inicialmente habilitado com permissões do usuário de sistema "*shell*"; que foi necessário realizar a instalação dos aplicativos "Screen Lock Bypass" e "Android Forensics Logical Application" para ignorar o sistema de controle de acesso; que foram extraídos dados de aplicativos em execução, fornecendo acesso de super usuário ao *shell* do modo de depuração, sendo estes gravados na estação pericial; que a extração do cartão de memória ocorreu sem falhas, gerando um código de integridade; que o cartão não foi substituído para evitar um desligamento não desejável do sistema, sendo usado para armazenar os dados extraídos do telefone; que foram espelhadas as partições do sistema, cujo sistema de arquivos é o YAFFS2, por meio do comando *dd* e gerados os respectivos códigos de integridade; que, finalizando, foi utilizado o software "Android Forensics Logical Application" para extração dos dados de uma forma mais amigável, realizando também extração complementar por meio de inspeção visual, de forma manual por meio da navegação pelo sistema. As informações extraídas estão descritas na seção 5.4.2, que trata da realização dos exames.

Adicionalmente, no processo de documentação deve-se anotar as contas configuradas no dispositivo, assim como informar que se encontrava com sincronização automática. Também são registradas informações a respeito do telefone, como versão do Android, 2.2.2; versão do *kernel*, 2.6.32.9wfp018@zbr05lnxdroid03 #1; o número da versão, ShadowMOD-BR v0.9.16; o IMEI, 356698030331558; endereço MAC do Wi-Fi, 00:26:ba:17:db:b2; e desde quando se encontra ligado, 23/04/2011 22:43:00 (horário de Brasília).

5.4.2. Exame

Após definido que o objetivo dos exames é a extração completa das informações do telefone celular, a critério do analista pericial, busca-se individualizar o *smartphone* (processo f.2 da Figura 4.7).

Neste cenário, a individualização do telefone ocorreu já no momento da extração, quando foi possível obter a conta configurada no dispositivo, andmor1lima@gmail.com[30], que pode apresentar informações a respeito do nome do proprietário. Outra informação obtida no momento da aquisição dos dados é a referente ao aplicativo "Latitude", que apresentou a localidade do proprietário do *smartphone* e confirmou seu nome como André Morum. A seguir, passou-se para a análise do cartão de memória (processo f.3 da Figura 4.7).

A imagem do cartão de memória, "cenario3.dd", foi adicionada como evidência na ferramenta Forensic ToolKit (FTK, versão 1.81), para fins de facilitar sua análise. Desta forma, foi possível categorizar as imagens, documentos, vídeos, arquivos de texto, dentre outros, disponíveis no cartão de memória, onde foram filtradas as informações que pudessem ser úteis à investigação.

Com relação aos dados extraídos do telefone (processo f.4 da Figura 4.7), as imagens (cópias integrais) das partições *system*, *cache* e *userdata*, podem ser analisadas no FTK (AccessData, 2011) apenas com a opção de *data carving*, uma vez que não há o suporte ao YAFFS2, o que limita o exame. Uma opção ao examinador é usar uma estação pericial com o sistema operacional GNU/Linux com um *kernel* compilado para dar suporte a este tipo de sistema de arquivos, usando ferramentas forenses para este ambiente ao realizar o exame. Outra opção é, no momento da aquisição dos dados, realizar a extração lógica dos arquivos mais importantes para o exame, a exemplo do que foi realizado nesse cenário, logo após o espelhamento das partições do telefone, quando foram copiados os arquivos relativos às aplicações (inclusive os bancos de dados) e os arquivos de configuração do sistema (Lessard e Kessler, 2010).

No diretório */data/system*, foi possível obter a lista dos aplicativos instalados no sistema (arquivo "package.list") e a conta Google configurada para o telefone com sua senha cifrada (arquivo "accounts.db"). Na pasta */data/misc*, foi encontrada a rede Wi-Fi configurada e sua respectiva senha WPA2 armazenada em claro no arquivo "wpa_supplicant.conf".

Examinando os arquivos de *cache* obtidos a partir do diretório */data/data*, foram encontrados comprovantes de pagamento e transferência realizados, extrato da conta corrente e limites do cartão de crédito, referentes ao aplicativo "br.com.bb.android", conforme ilustrado na Figura 5.21. Foi possível verificar que no telefone havia instalado o aplicativo "Seek Droid" ("org.gtmedia.seekdroid"), que permite a localização, bloqueio, e deleção dos dados

[30] O e-mail foi alterado a fim de preservar o endereço verdadeiro.

remotamente a partir do sítio da Internet www.seekdroid.com. No diretório de instalação do aplicativo, foi encontrado o arquivo "prefs.xml" que continha o nome de usuário do aplicativo (dedemor), a senha de acesso (@#senhasecreta) e suas configurações, mostrando que o aplicativo estava autorizado a apagar os dados do cartão do telefone. O aplicativo "Gtalk", forneceu no arquivo "talk.db" históricos de bate-papo e lista de amigos; informações sobre envio e recebimento de e-mails com horários, remetente e destinatário foram obtidas a partir do arquivo "mailstore.andmor1lima@gmail.com.db" do aplicativo "com.google.android.gm"; mensagens SMS encontravam-se armazenadas no arquivo "mmssms.db" do aplicativo "com.android.providers.telephony"; já os eventos de calendário foram encontrados no arquivo "calendar.db" do aplicativo "com.android.providers.calendar"; a partir do arquivo "webview.db", do aplicativo "com.android.browser", constatou-se que o usuário do telefone se autenticou nos sítios do Facebook (http://m.facebook.com), webmail do DPF (https://webmail.dpf.gov.br), Yahoo (http://m.login.yahoo) e Mercado Livre (https://www.mercadolivre.com); já no arquivo "DropboxAccountPrefs.xml" do aplicativo "com.dropbox.android", foi possível obter o nome do usuário configurado (mor1lima@yahoo.com), assim como o arquivo "db.db" possuía a listagem dos diretórios e arquivos, com seus respectivos tamanhos. Já as configurações do sistema foram encontradas no arquivo "settings.db", referente ao aplicativo "com.android.provider.settings". Muitas outras informações podem ser obtidas a partir dos arquivos de *cache* e bancos de dados dos aplicativos, devendo o analista pericial aprofundar o exame a medida da necessidade de se atingir o objetivo.

Com a finalidade de complementar o exame, foram analisados os relatórios gerados a partir da ferramenta forense "Android Forensics Logical Application", da Via Forensics. Foram gerados arquivos ".csv", analisados a partir de um editor de planilhas. Foi possível obter os contatos da agenda telefônica e mensagens SMS. Além disso, foram obtidos 147 registros de ligações recebidas, realizadas e não atendidas, o histórico de navegação web e os e-mails das pessoas com as quais foram trocadas mensagens eletrônicas. Com relação ao calendário, a partir da extração manual dos registros, foi possível observar que o proprietário ia ao dentista uma vez por mês, realizava tratamento de alergia, supostamente foi ao Terraço Shopping nos dias 3, 10 e 17 de abril de 2011, as 17hs, que no dia 7 de abril de 2011 às 09h30min tinha um encontro com orientador do mestrado na UnB, dentre vários outros compromissos agendados.

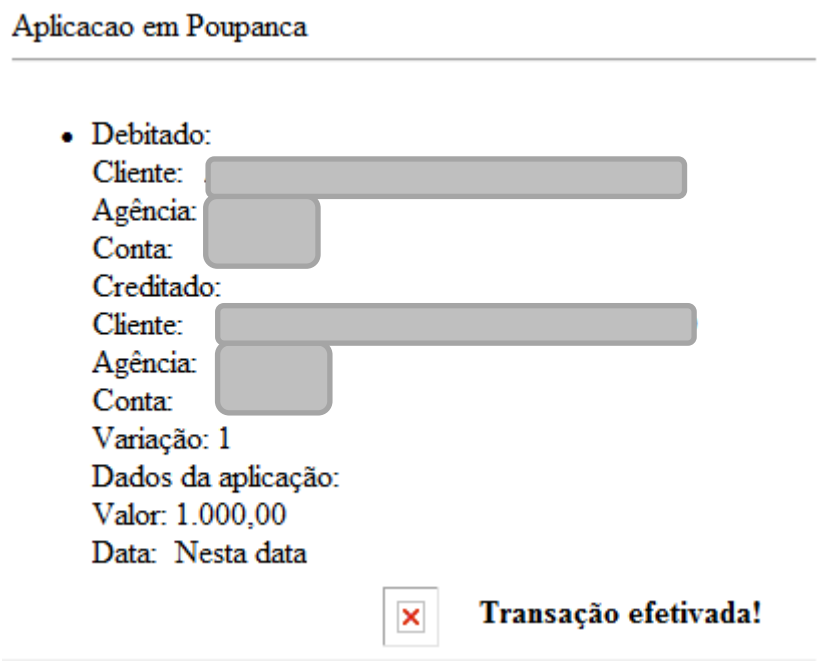


Figura 5.21 - Comprovante de aplicação em poupança.

Os dados extraídos a partir da ferramenta forense e visualizados diretamente no telefone estavam conformes com os obtidos a partir da análise do diretório */data/data*, extraídos diretamente da memória do telefone. Não houve necessidade de realizar intervenções manuais no telefone para complementar os dados obtidos a partir da fase de extração.

Com relação aos arquivos de *dump* de memória, usou-se um editor hexadecimal para examiná-los. Outra opção é inseri-los como evidência no FTK (AccessData, 2011) e processá-los, uma vez que esta ferramenta possui um visualizador hexa e é possível realizar filtragem de *strings*. Buscou-se informações como login e senha nestes arquivos, entretanto sem sucesso.

Conforme citado, foi encontrado no *smartphone* o aplicativo "Dropbox", que funciona como um repositório de arquivos na Internet (computação em nuvem). Assim, pode ser interessante avaliar junto com a autoridade solicitante da perícia a necessidade de obter os arquivos armazenados neste repositório (f.5 e f.6). Também pode ser interessante ter acesso à conta cadastrada no telefone, com acesso irrestrito a caixa de e-mail, calendário, contatos e outros serviços associados. A Figura 5.22 ilustra a etapa de exame do cenário descrito.

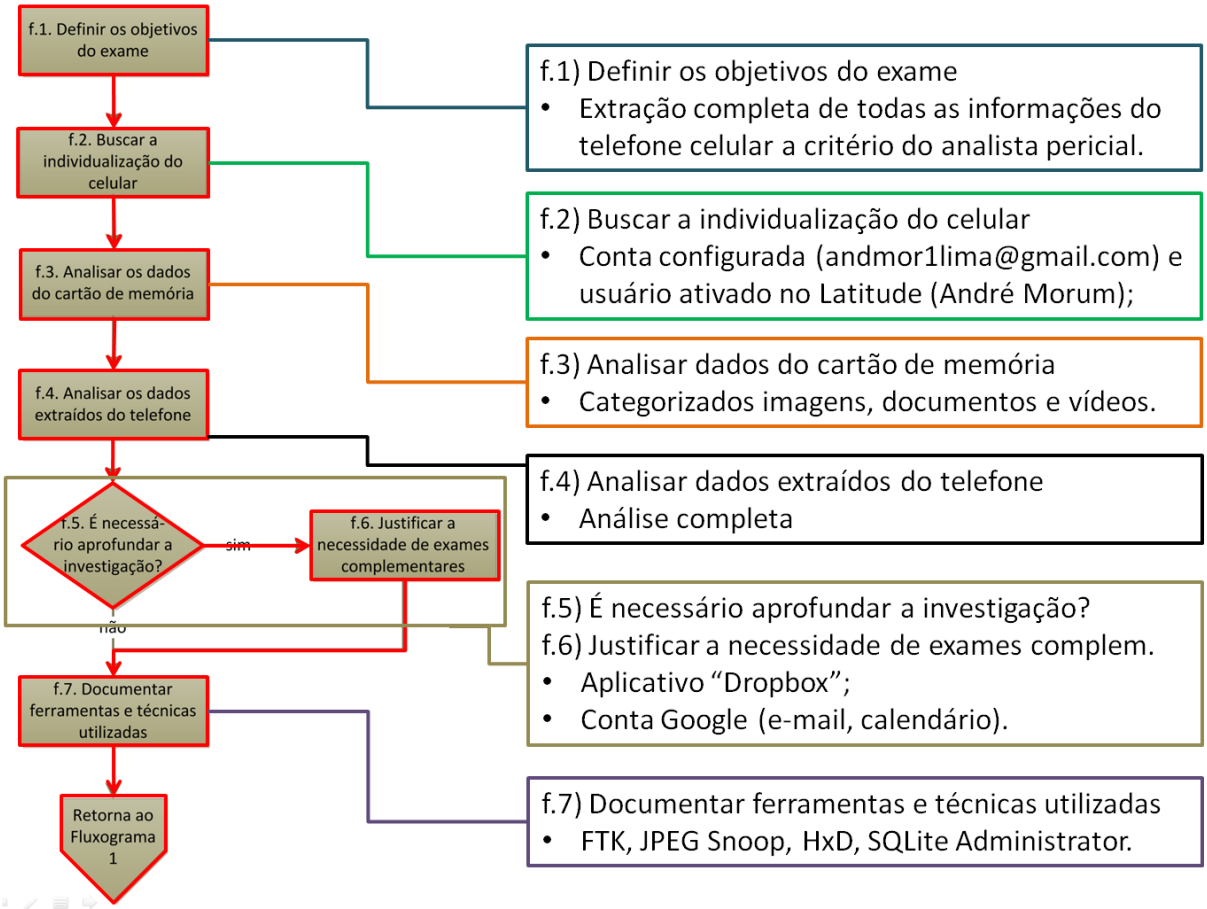


Figura 5.22 – Etapa de exame do cenário 3 (Motorola Milestone).

A fim de documentar as ferramentas e técnicas utilizadas (processo f.7 da Figura 4.7), é necessário esclarecer que para realização do exame, foi utilizada a ferramenta Forensic Toolkit da Access Data, versão 1.81, que possui um visualizador hexadecimal e sistema de busca indexada por palavras-chave; um editor hexadecimal (HxD, versão 1.7.7.0[31]); a ferramenta SQLite Administrator, versão 0.8.3.2 beta, para análise dos bancos de dados; e a ferramenta JPEGSnoop, versão 1.5, para extração dos metadados das fotos. No cartão de memória foi possível buscar os dados apagados que puderam ser recuperados por meio da técnica de *carving*, categorizando os tipos de arquivos encontrados na mídia.

5.5. O LAUDO/RELATÓRIO PERICIAL

Nos cenários propostos, os laudos/relatórios periciais não foram gerados de fato. Cada instituição possui seu modelo para documentar os processos relativos a análises periciais. Desta forma, é importante ao analista, independente do cenário proposto, documentar o fluxo

[31] http://www.mh-nexus.de, acessado em 26/04/2011.

percorrido quando da análise do *smartphone*, descrevendo os procedimentos realizados, citando as ferramentas utilizadas, justificando e fundamentando as ações tomadas na análise do sistema Android.

É importante o documento pericial conter claramente a descrição do dispositivo encaminhado, o objetivo do exame, um breve histórico sobre a apreensão, quando for o caso, como se procedeu à extração dos dados; descrevendo as técnicas utilizadas, quais ferramentas e procedimentos usados para a análise dos dados extraídos e; finalmente, uma conclusão clara e objetiva daquilo que se estava buscando com a devida fundamentação técnica para se chegar àquele resultado.

5.6. ANÁLISE DOS RESULTADOS

O método proposto foi testado por meio de sua aplicação no exame de três *smartphones* Android que abordavam diferentes situações que um analista pode se deparar, como exposto nos cenários apresentados e resumidos na Tabela 5.2. Os dois primeiros cenários buscaram simular a situação real com que os celulares são apreendidos e encaminhados aos setores periciais na Polícia Federal. Já o terceiro cenário buscou uma das situações mais complexas com que o analista pericial pode enfrentar. Os três cenários apresentaram resultados satisfatórios da análise dos *smartphones*, em que foram extraídas de informações do dispositivo, resguardando e documentando as evidências processadas.

Tabela 5.2 - Cenários utilizados para validar o método proposto.

	Apreensão		Aquisição					Exame
	Analista no local	Suspeito no local	Ligado	Cartão Removível	Bloqueado	Desblo-queável	Super usuário	Exames adicionais
Cenário 1	Não	Não	Sim	Não	Não	Não se aplica	Não	Não
Cenário 2	Não	Não	Não	Sim	Sim	Não	Prová-vel	Sim*
Cenário 3	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim**

* Exames complementares para tentar ter acesso ao aparelho.
** Exames complementares nos aplicativos que usam computação em nuvem.

Os resultados obtidos a partir do método proposto foram importantes, dado que nos modelos atualmente propostos de análise forense em telefones celulares, pouco se discute as peculiaridades de cada plataforma. A partir da especificação de um método para a plataforma Android, foram mapeadas as dificuldades encontradas pelos analistas periciais, preparando-os e auxiliando-os a realizar uma análise da evidência, evitando imprevistos no decorrer do processo pericial, dada as diversas situações com que podem se deparar, o que poderia acarretar a perda de provas materiais.

6. CONCLUSÕES

Com os conhecimentos necessários a respeito de forense em telefones celulares e da plataforma Android, foi apresentado um método capaz de auxiliar o perito em tecnologia da informação desde o momento da apreensão do *smartphone* até a geração do relatório/laudo pericial.

A partir da diagramação dos processos e decisões que o analista pericial pode se deparar, foram mapeadas as situações que ele encontrará no decorrer de uma análise de um *smartphone* com o sistema Android. Assim, foi apresentado de forma clara e objetiva um método eficiente para a extração das informações armazenadas no equipamento por meio da utilização de procedimentos e técnicas forenses aplicáveis na análise de dispositivos móveis.

Com o conhecimento adquirido a partir das técnicas descritas em abordagens utilizadas internacionalmente, agregando a elas as especificidades da plataforma Android, foi possível descrever como o perito pode atuar ao se deparar com um *smartphone*, com sistema de controle de acesso, ou com cartões de memória embutidos, ou acesso de super usuário, e até mesmo como ele pode utilizar o modo de depuração USB para obtenção dos vestígios armazenados no dispositivo.

Mais especificamente, a partir do estudo das abordagens existentes de análise em telefones celulares e das características da plataforma Android, além de se propor um método que identificou as diferentes situações que um analista pericial enfrentará em um exame de um *smartphone* com sistema Android, foi possível: extrair as informações presentes no aparelho, dada cada situação apresentada; preservar da forma mais adequada os dados presentes no celular; documentar os procedimentos e técnicas utilizados, justificando sua aplicação e; consequentemente, fornecer ao processo investigativo evidências obtidas de forma incontestável a partir do dispositivo analisado.

O método proposto definiu um *workflow* flexível, capaz de adequar as diferentes técnicas e abordagens ao fato investigado. Assim, o analista pericial pode suprir a equipe de investigação das informações com as informações que podem ser imediatamente extraídas e analisadas, podendo informar a necessidade da realização de outros exames, ou até mesmo da possibilidade de extrair mais informação do *smartphone* apreendido a partir de técnicas mais invasivas de acesso ao sistema.

Diferentemente do que há publicado sobre análise pericial em *smartphones*, o método proposto atua de forma a abordar todos os processos que envolvem o tratamento do *smartphone* Android como uma evidência. Enquanto na literatura atual há uma ênfase na descrição de técnicas específicas para extração da informação dos dispositivos móveis com sistema operacional Android e abordagens genéricas e amplas para analise forense em telefones celulares, a partir dos testes realizados nos estudo de casos, observou-se que o método proposto nesse trabalho fornece ao analista pericial uma visão geral no tratamento de um dispositivo móvel com o sistema operacional Android, e, ao mesmo tempo, considerar as peculiaridades da plataforma (controle de acesso, cartões de memória embutidos, acesso de super usuário, modo de depuração e aplicativos de acesso remoto) de maneira a descrever técnicas e procedimentos para auxiliá-lo no decorrer do processo forense.

6.1. TRABALHOS FUTUROS

A partir do que foi estudado nesta dissertação e com a finalidade de dar continuidade ao trabalho desenvolvido seria interessante aperfeiçoar as técnicas de extração dos dados armazenados no *smartphone* por meio do desenvolvimento de métodos, ainda mais específicos, que consigam extrair a maior quantidade de informação do sistema operacional, dada sua versão e sistema de arquivos, assim como aplicações mais eficientes para este fim.

Outro trabalho interessante de se desenvolver seria uma ferramenta, ainda não disponível, para exame forense com suporte ao sistema de arquivos YAFFS2. A criação desta ferramenta pericial poderia facilitar a fase de extração dos dados armazenados na memória interna do telefone, pois não haveria necessidade do analista se preocupar com a forma que os dados serão examinados a partir de uma imagem gerado do sistema. Assim, facilitaria a fase do exame, uma vez que seria possível acessar as imagens geradas do sistema de forma similar ao que é realizado atualmente com imagens de discos rígidos formatados com os sistemas de arquivos mais utilizados (FAT32, NTFS, EXT3).

Outro trabalho que poderia ser desenvolvido seria de avaliar o método proposto neste trabalho ou, se for o caso, propor um específico, para análise pericial de dispositivos do tipo *Tablet* PC com o Android 3.x, tendo em vista a crescente presença deste equipamento no mercado mundial.

Com base em informações presentes neste trabalho, foi submetido um artigo intitulado “Aquisição de Evidências Digitais em *Smartphones* Android” à ICoFCS 2011 (The Sixth

International Conference on Forensic Computer Science), que foi publicado nos anais dessa conferência e apresentado na VIII Conferência Internacional de Perícias em Crimes Cibernéticos (ICCyber 2011) no período de 05 a 07 de outubro de 2011.

REFERÊNCIAS BIBLIOGRÁFICAS

AccessData. Forensic Toolkit (FTK) Computer Forensics Software. Sítio da internet da AccessData, 2011. Disponivel em: <http://accessdata.com/products/computer-forensics/ftk>. Acesso em: 10 outubro 2011.

Anatel. Brasil fecha maio com mais de 215 milhões de acessos móveis. Sítio da Agencia Nacional de Telecomunicações, 2011. Disponivel em: <http://www.anatel.gov.br/Portal/exibirPortalPaginaEspecialPesquisa.do?acao=&tipoConteudoHtml=1&codNoticia=22917>. Acesso em: 08 ago. 2011.

Antonioli, D.; Pilz, M. Analysis of the Java Class Format. [S.l.]. 2008. (98.4).

Ashcroft, J. Eletronic Crime Scene Investigation: A Guide for First Responders U.S. Department of Justice. DoJ. Washington, DC, p. 82. 2001.

Association of Chief Police Officers. Good Practice Guide for Computer-Based Electronic Evidence - Versão 4.0. [S.l.]. 2008.

Bahareth, M. iSay - Kings of the Internet. 1a. ed. [S.l.]: Trafford, 2010.

Bornstein, D. Dalvik VM Internals. 2008 Google I/O Session, 2008. Disponivel em: <http://sites.google.com/site/io/dalvik-vm-internals>. Acesso em: 12 abril 2011.

Brady, P. Anatomy & Physiology of an Android. 2008 Google I/O Session, 2008. Disponivel em: <http://sites.google.com/site/io/anatomy--physiology-of-an-android>. Acesso em: 12 abril 2011.

Brasil. Lei no. 3.189, de 3 de outubro de 1941, alterada pela lei 10.695, de 1 de julho de 2003. Código de Processo Penal, artigos 530-C e 530-D. Brasília: [s.n.], 2003.

Burnette, E. Hello, Android. [S.l.]: Pragmatic Bookshelf, 2008. ISBN 978-1-934356-17-3.

Cannon, T. Android Reverse Engineering. Thomas Cannon, 2010. Disponivel em: <http://thomascannon.net/projects/android-reversing/>. Acesso em: 23 março 2011.

Cannon, T. Android Lock Screen Bypass. Thomas Cannon, 2011. Disponivel em: <http://thomascannon.net/blog/2011/02/android-lock-screen-bypass/>. Acesso em: 23 março 2011.

Cassavo, L. In Pictures: A History of Cell Phones. PCWorld, 7 maio 2007. Disponivel em: <http://www.pcworld.com/article/131450/in_pictures_a_history_of_cell_phones.html>. Acesso em: 22 março 2011.

Cavaleiro, D. Nokia e Microsoft confirmam parceria para enfrentar Apple e Google. Jornal de Negócios, 11 fevereiro 2011. Disponivel em: <http://www.jornaldenegocios.pt/home.php?template=SHOWNEWS_V2&id=468017>. Acesso em: 22 março 2011.

Cellebrite. Mobile Forensics and Data transfer solutions. Cellebrite, 2011. Disponivel em: <http://www.cellebrite.com/forensic-products/forensic-products.html?loc=seg>. Acesso em: 17 agosto 2011.

DITEC/DPF. Instrução Técnica no. 003/2010-DITEC. Dispõe sobre a definição de diretrizes e a padronização de procedimentos no âmbito das perícias de Informática na Polícia Federal., Brasília, 11 mar. 2010.

Ehringer, D. The Dalvik Virtual Machine Architecture. David Ehringer, março 2008. Disponivel em: <http://davidehringer.com/software/android/The_Dalvik_Virtual_Machine.pdf>. Acesso em: 17 fevereiro 2011.

Farley, T. Mobile Telephone History. Telektronikk, v. 3, p. 22 a 34, abril 2005.

Gadhavi, B. Analysis of the Emerging Android Market. The Faculty of the Department of General Engineering, San Jose State University. [S.l.], p. 88. 2010.

Google Inc. Android Debug Bridge. Android Developers, 2011a. Disponivel em: <http://developer.android.com/guide/developing/tools/adb.html>. Acesso em: 4 abril 2011.

Google Inc. Android Fundamentals. Android Developers, 2011b. Disponivel em: <http://developer.android.com/guide/topics/fundamentals.html>. Acesso em: 17 março 2011.

Google Inc. Android SDK. Android Developers, 2011c. Disponivel em: <http://developer.android.com/sdk/index.html>. Acesso em: 29 abril 2011.

Google Inc. Compatibility Program Overview. Android Open Source Project, 2011d. Disponivel em: <http://source.android.com/compatibility/overview.html>. Acesso em: 22 abril 2011.

Google Inc. Logcat. Android Developers, 2011e. Disponivel em: <http://developer.android.com/guide/developing/tools/logcat.html>. Acesso em: 4 abril 2011.

Google Inc. What is Android? Android Developers, 2011f. Disponivel em: <http://developer.android.com/guide/basics/what-is-android.html>. Acesso em: 8 abril 2011.

Hallinan, C. Embedded Linux Primer: A Practical Real-World Approach. 2a. ed. [S.l.]: Prentice Hall, 2010.

Hashimi, S.; Komatineni, S.; Maclean, D. Pro Android 2. 1a edição. ed. [S.l.]: Apress, 2010. ISBN 978-1-4302-2659-8.

Hoog, A. Android Forensics. Via Forensics, 29 maio 2009. Disponivel em: <http://viaforensics.com/wpinstall/wp-content/uploads/2009/08/Android-Forensics-Andrew-Hoog-viaForensics.pdf>. Acesso em: 26 novembro 2010.

Hoog, A. Android Forensics Logical Application (LE Restricted). Via Forensics, 2010. Disponivel em: <http://viaforensics.com/wiki/doku.php?id=aflogical:start>. Acesso em: 22 abril 2011.

Hoog, A. Google Maps Navigation - com.google.apps.maps. Via Forensics, 2010. Disponivel em: <http://viaforensics.com/wiki/doku.php?id=aflogical:com.google.android.apps.maps>. Acesso em: 20 abril 2011.

Hoog, A. Android Forensics - Investigation, Analisys and Mobile Security for Google Android. 1a. ed. [S.l.]: Syngress, 2011.

Jansen, W.; Ayers, R. Guidelines on Cell Phone Forensics - Recomendations of the National Institute of Standards and Technology. [S.l.]. 2007.

Khan, S. et al. Analysis of Dalvik Virtual Machine and Class Path Library. Security Engineering Research Group, Institute of Management Sciences. Peshawar, Pakistan, p. 33. 2009.

Knijff, R. V. D. Handbook of Computer Crime Investigation, Chapter 11 Embedded Systems Analysis. [S.l.]: Academic Press, 2001.

Lessard, J.; Kessler, G. C. Android Forensics: Simplifying Cell Phone Examinations. Small Scale Digital Device Forensics Journal, setembro 2010.

linux-mtd.infradead.org. General MTD documentation. Memory Technology Devices, 14 outubro 2008. Disponivel em: <http://www.linux-mtd.infradead.org/doc/general.html>. Acesso em: 12 abril 2011.

Lopes, M.; Gabriel, M. M.; Bareta, G. M. S. Cadeia de Custódia: Uma Abordagem Preliminar. Curitiba: Departamento de Medicina Forense e Psiquiatria, Universidade Federal do Paraná. 2007.

Netherlands Forensic Institute. Mobile Phone Forensics. Netherlands Forensisch Instituut, 2007. Disponivel em: <http://www.holmes.nl/MPF/>. Acesso em: 12 novembro 2010.

Owen, P.; Thomas, P.; Mcphee, D. An Analysis of the Digital Forensic Examination of Mobile Phones. [S.l.]: IEEE. 2010.

Paula, Y. Android, playing with dumpsys. Softteco, 5 abril 2011. Disponivel em: <http://softteco.blogspot.com/2011/04/android-playing-with-dumpsys.html>. Acesso em: 12 abril 2011.

Pettey, C.; Stevens, H. Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012. Gartner, 7 abril 2011.

Disponivel em: <http://www.gartner.com/it/page.jsp?id=1622614>. Acesso em: 8 abril 2011.

Pettey, C.; Tudor, B. Gartner Says Android to Become No 2 Worldwide Mobile Operating System in 2010 and Challenge Symbian for No 1 Position by 2014. Gartner, 10 setembro 2010. Disponivel em: <http://www.gartner.com/it/page.jsp?id=1434613>. Acesso em: 21 setembro 2010.

Quirke, J. Security in the GSM system. AusMobile. [S.l.], p. 26. 2004.

Rossi, M. Internal Forensic Acquisition for Mobile Equipments, n. IEEE, 2008.

Speckmann, B. The Android mobile platform. [S.l.]: Eastern Michigan University, Department of Computer Science, 2008.

SQLite. About SQLite. SQLite, 2011. Disponivel em: <http://www.sqlite.org/about.html>. Acesso em: 5 abril 2011.