UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITARIO DE CASTANHAL
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ANTHONY JEAN SOARES

COMPUTAÇÃO FORENSE EM DISPOSITIVOS COM SISTEMA OPERACIONAL ANDROID

CASTANHAL
2018

ANTHONY JEAN SOARES

COMPUTAÇÃO FORENSE EM DISPOSITIVOS COM SISTEMA OPERACIONAL ANDROID

Trabalho de conclusão de Curso apresentado para obtenção do grau de Bacharelado em Sistemas de Informação, Faculdade de Computação do Campus Universitário de Castanhal, Universidade Federal do Pará.
Orientador: Prof. Dr. José Jailton

CASTANHAL

2018

ANTHONY JEAN SOARES

COMPUTAÇÃO FORENSE EM DISPOSITIVOS COM SISTEMA OPERACIONAL ANDROID

Trabalho de conclusão de Curso apresentado para obtenção do grau de Bacharelado em Sistemas de Informação, Faculdade de Computação do Campus Universitário de Castanhal, Universidade Federal do Pará.
Orientador: Prof. Dr. José Jailton

Data de aprovação: 20/12/2018

BANCA EXAMINADORA:

_____
Prof. Dr. José Jailton
Orientador - UFPA

_____
Prof. Me. Jorge Cardoso
Eminador interno - UFPA

_____
Prof. Me. Felipe Brito
Examinador interno - UFPA

Dedico este trabalho aos meus pais, Janil e Silva Soares e José Francisco Carvalho Aguiar e minha esposa Mônica e filha, pelo amor, carinho e incentivo que me deram durante toda a minha vida pessoal e acadêmica, que sem dúvida contribuiu diretamente para minha formação pessoal e profissional.

A meus amigos que me deram suporte nos momentos que precisei, especialmente Clebson, Felipe, André, pelo apoio e paciência que recebi durante o início desse trabalho.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela vida, mesmo quando ela se monstra confusão, agradeço pela força e persistência, para realizar este trabalho, pois sem ele nada seria possível, agradeço a minha mãe, minha esposa Mônica e filha, pela capacidade de perdoar.

Agradeço meus amigos, Darlan e Clebson, pela companhia nas horas vagas tentando entender o processo de criação de “The Final Cut”.

Agradeço também, A Louis Amistrong, que mesmo depois de morto, tem me inspirado com suas canções e me dado paciência. Salve “La vie em rose”.

Aos meus irmãos, Yuri Nery Soares, Jusselino Andrey Soares, Jamily Soares, Marcela Aguiar, por existirem na minha vida e pela contribuição direta e indireta.

E em especial minha esposa, só nós sabemos o que passamos juntos para que eu pudesse chegar até o fim deste curso.

Aos meus primos, que sempre me incentivaram, em especial Welton Igor, por todos os momentos de descontração que passamos juntos. Aqui meu muito obrigado!

A todos os meus Professores, que durante a minha formação, contribuíram de maneira que eu não tenho palavras para descrever a importância. Vocês com certeza fazem parte desta jornada.

Ao meu orientador Prof. Dr. José Jailton Junior, por todas as contribuições e dedicação durante a realização deste trabalho por mim logrado.

A todos, o meu muito obrigado!

“Não faças nunca depender a tua felicidade de algo que não dependa de ti”.
Huberto Rohden

RESUMO

Na atualidade, vivemos em uma sociedade cada vez mais global digital e dependente de novas tecnologias que nos permitem estar conectadas a qualquer parte do mundo, junto a essas novas tecnologias vislumbramos os smartphones, que agregam diversos recursos e serviços disponibilizados por operadoras de telefonia móvel. Milhares de pessoas perceberam a grande utilidade destes dispositivos e estão fazendo uso dos smartphones com o sistema operacional Android. Esta utilização em larga escala destes dispositivos alimenta o imenso repositório de informações pessoais. Desta forma, a perícia computacional busca coletar provas e evidências digitais em dispositivos móveis e vem evoluindo para aplicar técnicas específicas nestes aparelhos celulares.

Portanto, neste trabalho será utilizado um método de extração de informações e coleta de dados para a análise destes dispositivos móveis, utilizado o sistema operacional *Cyborg Hawk 1.1*, do qual possui um modulo específico para análise de forense em smartphones.

Palavras-chave: Pericia Forense; Sistema Operacional Cyborg Hawk, Sistema Operacional Android, Android SDK Manager e AFLogical-OSE.

ABSTRACT

In the updated, we live in an increasingly global digital society and dependent on new technologies that allow us to be connected to any part of the world, along with these new technologies we see the smartphones, which add various features and services provided by mobile operators. Thousands of people have realized the great utility of these devices and are making use of smartphones with the Android operating system. This large-scale use of these devices feeds the huge repository of personal information. In this way, the computational expertise seeks to collect evidence and digital evidence in mobile devices and has been evolving to apply specific techniques in these cellular devices.

Therefore, in this work will be used a method of extracting information and data collection for the analysis of these mobile devices, using the operating system "Cyborg Hawk", which has a specific module for forensics analysis in smartphones.

Keywords: Forensic expertise; Cyborg Hawk Operating System, Android Operating System, Android SDK Manager and AFLogical-OSE.

LISTA DE FIGURAS

LISTAS DE QUADROS

LISTA DE ABREVIATURAS

ADB	Android Debug Bridge
ADBD	Android Debug Bridge Daemon
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CPP	Código do Processo Penal
CPU	Central Processing Unit
ENIAC	Eletronic Numerical Interpreter and Computer
EXIF	Exchangeable Image File
IBM	International Business Machines
IrDA	Infrared Data Association
IRS	Internal Revenue Service
IPC	Interprocess communication
FLETC	Federal Law Training Center
Dalvik VM	*Dalvik Virtual Machine*
FGV	Fundação Getúlio Vargas
GNU GPL	GNU General Public License
GPS	Global Positioning System
HTC	High-Tech Computer
HTML	HyperText Markup Language
IBGE	Instituto Brasileiro de Geografia e Estatística
ISFS	Information Security and Forensics
JDK	Java Development Kit
JVM	Java Virtual Machine
MMS	Multimedia Mensaging Service
NDK	Native Developmente Kit
NIST	National Institute of Standards and Tecnology
OOB	Out-Of-Band
OHA	Open Handset Alliance
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PUK	PIN Unlock Key
RAM	Random Access Memory
ROM	Read-Only Memory
RCMP	Royal Canadian Mounted Police
SD	Secure Digital Card
ADB	Android Debug Bridge
ADBD	Android Debug Bridge Daemon
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CPP	Código do Processo Penal
CPU	Central Processing Unit
ENIAC	Eletronic Numerical Interpreter and Computer
IBM	International Business Machines
SDK	Software Development Kit

SIM	Subscriber Identity Module
SMS	Short Mensage Server
SSLs	Secure Socket Layers
USB	Universal Serial Bus
VM	Virtual Machine
WIFI	Wireless Fidelity
YAFFS2	Yet Another flash File System

SUMÁRIO

1 INTRODUÇÃO

Nas últimas décadas é irrefutável o aumento do uso de *smartphones* pela população. Conforme pesquisa realizada pelo IBGE[1] (apud Gomes, 2018), O Brasil finalizou o ano de 2016 com 116 milhões de pessoas conectadas à internet, o equivalente a 64,7% da população com idade acima de dez ano.

A Pesquisa mostra também, que os *smartphones* continuam sendo o principal aparelho usado para acessar a internet no Brasil e que em 2016 o dispositivo eletrônico é usado por 94,6% das pessoas, à frente de dispositivos computadores (63,7%), tabletes (16,4%) e televisão (11,3%). E, segundo o IBGE, 77,1% dos brasileiros possuíam algum tipo de celular. Uma outra pesquisa realizada em 2018 pela FGV (apud Demartini, 2018), afirma que existe mais smartphones ativos por pessoas e que hoje há 220 milhões de smartphones em funcionamento, contra 207,6 milhões de habitantes.

Os smartphones são formas compactas de computadores de alta performance[2], grande capacidade de armazenamento, e por sofrerem grandes melhorias constantemente, além de terem evoluídos muito em relação a versões anteriores, são considerados dispositivos portáteis mais pessoais que o usuário pode possuir.

Eles são usados para realizar tarefas de comunicação, como envio de mensagem de texto e ligações, tarefas que demandam mais tecnologia como navegação na web, uso de aplicativos de e-mail, capturas de imagens e vídeos; criação e armazenamento de documentação, e identificação de localidades com os serviços de GPS[3], dentre outros.

1.1 Justificativa

Os smartphones são aparelhos que possuem diversas funcionalidades, processam uma quantidade enorme de informações de seus usuários, revolucionaram a maneiras como a sociedade se comunica e se comporta. Esse avanço proporcionou um desafio para a computação forense, que precisou desenvolver-se e criar técnicas e métodos, para avaliar as informações que são armazenas nos dispositivos móveis, como smartphones.

---

[1] IBGE Instituto Brasileiro de Geografia e Estatística.

[2] Por que estes dispositivos apresentam um grande poder computacional, oferecendo aos usuários uma grande diversidade de aplicações.

[3] GPS Instrumento de navegação por satélite criado para fins militares, hoje define a localização do dispositivo usando a rede de satélite GPS, também permite que aplicações utilizem o sistema.

Por tanto, nessa pesquisa será apresentado um método pericial voltado para smartphones com sistemas operacional Android, e com foco primordial na recuperação de mensagem SMS[4] e MMS[5], além de outras mídias encontradas no dispositivo. O presente trabalho também mostrará a importância da utilização das técnicas corretas e métodos homologados de analise pericial, aceitas por entidades profissionais, possibilitando a produção de conhecimento necessário para a condução de investigações forense voltadas para telefones celulares.

1.2 Objetivos

1.2.1 Objetivos gerais

O principal objetivo desse trabalho é demostrar técnicas e procedimentos que auxilie no processo de análise forense em dispositivos móveis, utilizado uma estação de trabalho com o Sistemas Operacional Linux e fermentas de código aberto, além de contribuir com o enriquecimento de conhecimentos técnico sobre a área de computação forense em dispositivos móveis com o sistema Android.

1.2.2 Objetivos especifico

A pesquisa tem como objetivo especifico:

a) Preparar uma estação de trabalho com Sistemas Operacional Linux utilizando ferramentas Open Soure;
b) Emular um dispositivo Android utilizando o *SDK Manager*...
a) Descrever técnicas de análise forense em dispositivos móveis com Android;
b) Mostrar como é o funcionamento da ferramenta *AFLogical-OSE_.5.2* da "*viaForensce"*.

1.3 Aspectos metodológicos

Para a concepção desta pesquisa optou-se primeiramente por realizar uma revisão de literatura sobre a evolução da computação forense e sua aplicada em telefones celulares, produzidas

[4] SMS (*Short Message Service)* é um recurso de rede celular que permite aos usuários enviar e receber mensagens de texto de até 160 caracteres alfanuméricos em seus aparelhos.
[5] MMS (*Multimídia Messaging Service*) é um padrão aceito para mensagens que permite aos usuários enviar e receber mensagens formatadas com texto, gráficos, fotografias, áudio e videoclipes.

por profissionais da área, foi realizado pesquisas na web para acrescentar informações complementares, através de livros, artigos.

Foi possível verificar que as abordagens de análise forense em computadores são diferentes das utilizadas em smartphones como Android, uma vez que esses dispositivos necessitam de interação no momento da análise.

Outro problema observado é que os estudos relacionados a computação forense em dispositivos móveis com sistema operacional Android, partem de característica gerais, devido à grande variedade de fabricante e modelos e sistemas de segurança e controle de acesso em detrimento de aspectos específicos de cada plataforma.

Por fim, foram estudados aspectos específicos de dispositivos móveis com a plataforma Android, mostrando características gerais do sistema, seus aplicativos e suas ferramentas sob um prisma forense, utilizado práticas atuais proposta pelo NIST, Policia Federal e outras instituições.

1.4 Estrutura do trabalho

Este trabalho tem como finalidade descrever um método para executar a análise forense com o sistema operacional Android, baseado em práticas utilizados por especialistas e desenvolvidas por profissionais da área.

Além da introdução do contexto historio e evolução da computação forense até sua aplicação em dispositivos móveis, e da apresentação do tema, sua justificativa e objetivos, o trabalho está estruturado da seguinte forma:

Capitulo 2: Apresentar conceitos relevantes sobre a computação forense, história da computação forense e sua evolução, apresentar um referencial teórico importante para sua compreensão e definição; mostrar quais os profissionais que estão evolvidos, e definir o que são evidência digitais.

Capitulo 3: Será abordado as melhores práticas para aquisição de dados em dispositivos móveis, o processo utilizado para busca, apreensão e conservação; aquisição dos dados, as ferramentas de extração, memorias dos dispositivos móveis, códigos de segurança e controle de acesso e memorias removíveis.

Capitulo 4: Apresenta uma visão geral sobre o sistema operacional *Android*, suas versões, arquiteturas. Além de mostrar a importância da Máquina *Virtual Dalvik* para a perícia forense em smartphones com Android, apresentar características do Android *Software Development Kit*, *Android Virtual Devices* e Depuração Via USB, sistemas de arquivos utilizados pelo sistema Android, partições do sistema de arquivo, arquivos de log do sistema e das aplicações.

Capitulo 5 apresenta um estudo de caso para a perícia forense em Android, utilizando software Android SDK (*Software Development Kit*) no Sistemas Operacional Cyborg Hawk 1.1, traz também especificações do ADB (Android Debug Bridge) e de sua utilização, modelo de segurança e permissões de super usuário.

Capitulo 6 Aborda o processo de instalação e configuração da Máquina Virtual, instalação do sistema operacional para a estação de trabalho, além de configurações especificas e atualização SDK Manager para criação do dispositivo Android virtualizado e por fim, a verificação e execução do Android Debug Bridge e ativação da Depuração UBS do dispositivo criado.

Capitulo 7 é apresentado a ferramenta AFLogical_OSE_1.5.2, seu processo de instalação, além da realização da simulação de um teste em um dispositivo Emulado no Android ADK Manager.

Capitulo 8 será apresentado um estudo de caso, utilizando as ferramentas proposta no seguinte senário, utilizando um aparelho ligado, desbloqueado e sem permissão de super usuário,

Capitulo 9 será apresentado um outro estudo de caso com aparelho ligado, sem bloqueio de tela padrão e com acesso a depuração USB ativada,

Por último no Capitulo 10 será apresentado um último estudo de caso, com um aparelho ligado com bloqueio de tela padrão e com acesso a depuração USB ativada.

Por fim, no Capítulo 11 será apresentado uma síntese do trabalho, descrevendo sua relevância, dificuldades obtidas a sim como mostrar a importância de se dar continuidade do trabalho e sua aplicação em outras plataformas.

2 CIÊNCIA FORENSE

A Ciência Forense pode ser definida como o conjunto de conhecimentos científicos e técnicas que podem ser utilizadas para desvendar e solucionar problemas e crimes e também diversos assuntos legais, tais como cíveis, criminais, penais ou ainda administrativos. Ela é também considerada uma área interdisciplinar, envolvendo diversas disciplinas como física, química, biologia, dentre outras. (MESQUITA, 2018).

De acordo com a definição concebida por Foltran e Shibatta (2011) a ciência forense é considerada como multidisciplinar pois utiliza vários elementos de outras ciências para que possa ser feita uma análise correta de um possível vestígio ou crimes. Dentre as áreas utilizadas pela ciência forense, destacam-se: a papiloscopia, a balística forense, genética forense, a computação forense.

A Ciência Forense, proporciona os princípios e técnicas que possam facilitar a investigação de determinado delito ou crime, de outra forma; a ciência forense é qualquer princípio ou técnica que possa ser usado para identificar, desvendar, recuperar, construir ou analisar uma evidencia durante uma investigação criminal. (FOLTRAN; SHIBATTA, 2011).

2.1 Computação forense

A Computação forense é a ciência que utiliza de artifícios e técnicas especializadas para tratar da coleta, preservação e análise de dados eletrônicos em um incidente computacional ou que envolvam a computação forense como prática ou meio de praticá-la.

Segundo Junior e Claro (2011, p. 7), a computação forense está inserida em um campo que se baseia na realização de procedimentos e exames periciais que objetiva a elucidação ou esclarecimento de práticas ilícitas ou delituosas, relacionadas a área de informática.

Em outras palavras, os exames e procedimentos realizados pela computação forense são aplicados em dispositivos de armazenamento computacional, site de internet, mensageiros eletrônicos (e-mail) e dispositivos movíeis. Estes exames objetivam analisar, identifica, coletar informações, localizar dispositivos invadidos ou invasores, que estejam de alguma forma relacionados a crimes e fraudes e que possam fornecer algum indício relevante para comprovar o ato de delitos relacionados a informática.

De acordo com Eleutério & Machado (2010. p. 16):

> "A Computação Forense tem como objetivo principal determinar à dinâmica, a materialidade e autoria de ilícitos ligados à área de informática, tendo como questão principal a identificação e o processamento de evidências digitais em provas materiais de crime, por meio de métodos técnico-científicos, conferindo-lhe validade probatória em juízo".

Portanto, o objetivo da análise em dispositivos computacionais é a extração de qualquer vestígio que possibilite a elaboração de matéria probante, relacionado ao caso investigado, legalmente aceito, que comprove a existência de um ilícito e sua autoria. (CLEMENTE, 2009, p. 20).

2.2 Breve história da evolução da computação forense

A computação avançada teve seu início na década de 1946, nos Estados Unidos, com o surgimento dos primeiros computadores eletrônicos, que se caracterizavam por possuírem dezenas de toneladas e milhares de componentes, como por exemplo o ENIAC[6], primeiro computador de grande porte totalmente eletrônico (NETO, 2009, p. 9).

Logo após a invenção do ENIAC, em 1947, surgiu a nova geração de computadores, que traziam como modernidade, a invenção dos transistores, que possibilitavam a essa nova geração de máquinas serem 100 vezes mais rápidos que os computadores que utilizavam válvulas. (NETO, 2009, p. 9).

Passados os anos e a computação evoluindo, por volta 1971, a técnica de fabricação de circuitos integrados começou a ser aperfeiçoada e a criação do primeiro software para microcomputadores foi desenvolvido. Neste período, a maioria dos computadores ainda eram *mainframes*[7], utilizados por pessoas, para trabalharem em finanças, engenharia e dentre outras organizações e ouve um aumento dos crimes eletrônicos, relacionados ao setor financeiro. Os policiais não conheciam o suficiente sobre computadores e não sabiam fazer as perguntas corretas para elucidar os crimes, ou para preservar as provas para o julgamento. Muitos, começaram a participar do FLETC (Centros Federais de Treinamento em Polícia**),** que se destinava em oferecer

[6] ENIAC (*Eletronic Numerical Interpreter and Computer*), foi o primeiro computador eletrônico de propósito geral. Ele foi projetado principalmente para calcular tabelas de disparo de artilharia a serem usadas pelo Laboratório de Pesquisa Balística do Exército dos Estados Unidos.

[7] Mainframes são computador de grande porte, dedicado normalmente ao processamento de um volume grande de informações.

treinamento na aplicação da lei da nação e, em áreas comuns a policiais e na recuperação de dados digitais. O crime mais conhecido nesta época foi "*One-half cent crime*[8]", que significa: "Crime de meio centavo". (NELSON; PHILLIPS; STEUART, 2010, p. 5).

Por volta de 1980, os computadores começaram a ganhar popularidade e diferentes sistemas operacionais começaram a surgir, como o *Apple 2E* em 1983 e o *Macintosh* em 1984, pela *Apple*. E, posteriormente surgiram sistemas operacionais de Disco (DOS), incluindo PC-DOS, Q-DOS, DR-DOS, IBM-DOS E BS-DOS. Nesta época, as ferramentas forenses eram simples e eram criadas por agências governamentais, não disponíveis ao público, tais como*:* RCMP (*Royal Canadian Mounted Police),* que detinha de suas próprias ferramentas de investigação e a IRS (*U.S. Internal Revenue Service).*

Em meados dos anos 80, surgiram no mercado o *Xtree Gold*, que reconhecia tipos de arquivos, recuperava arquivos perdidos ou apagados e permitia ver o código de um binário, convertendo seus bytes para tabela ASCII. Apareceu então, o *Norton DiskEdit*, ferramenta preferida para encontrar arquivos deletados. Era possível usar esta ferramenta nos PCs mais poderosos e modernos da época, como os PCs *IBM*, compatíveis com essa ferramenta, possuíam 10 MB de discos rígidos e duas unidades de disquetes, conforme figura 1 logo abaixo:

Figura 1 — Um computador 8088.

Fonte: NELSON, PHILLIPS, STEUART, 2010, p. 5.

[8] One-half cent crime era um crime onde os bancos costumavam rastrear o dinheiro em contas até a terceira casa decimal ou mais. Eles usaram e ainda usam o método contábil "arredondamento" quando pagam juros. Alguns programadores de computador corromperam esse método abrindo uma conta para si próprios e escrevendo programas que desviavam todas as verbas fracionárias para suas contas.

Em 1987, a *Apple* produzi-o o *Mac SE*, que era um *Macintosh*, com um disco rígido externo de 60MB. Nesta mesma época, o computador mais popular era o *Commodore 64*, ele ainda usava fitas de áudio padrão para gravar dados, de modo que o *Mac SE* representou um avanço importante na tecnologia de computadores.

2.3 Profissionais da perícia forense computacional

O Perito forense computacional é o profissional responsável por analisar ambientes digitais que compõem: software, hardware e mídias, afim de detectar fraudes e levantar evidência que possam servir como provas para solucionar crimes. O perito forense computacional, deve ter conhecimento em tecnologias da informação, deve estar em constante capacitação, para realizar as investigações e fazer levantamento das evidências.

Conforme Eleuterio e Machado (2011, p. 14), os profissionais que atuam nesta área devem ser treinados e conhecer ao máximo as tecnologias da informação, devem estar capacitados para o trabalho de investigação, levantamento e preservação das provas materiais e elaboração de laudo pericial.

No que concerne a elaboração do laudo pericial, os peritos descreveram minuciosamente o que eximirão e responderam ao que formularam.

Ainda segundo Eleuterio e Machado (2011, p. 16):

> "No caso especifico da Computação quem realiza esse trabalho de forma oficial no âmbito criminal é o perito criminal em informática. Entretanto, diversos outros profissionais podem ter a necessidade de realizar exames em computação. São eles: peritos particulares, auditores em sistemas, profissionais de TI e outros. Além disso, juízes, advogados e delegados, promotores e demais profissionais da área de direto, também devem conhecer como evidências e provas digitais devem ser corretamente coletadas, apuradas e apresentadas."

Portanto de acordo com os autores acima citados, o sucesso da perícia forense computacional é consequência do nível de conhecimento e qualificação adequada em tecnologias da informação, que possuem os profissionais envolvidos nas atividades de computação forense. Além da habilidade técnica em que se exige dessas profissionais, o sucesso da computação forense, também está diretamente relacionada a capacidade de entendimento no que concerne a coleta de provas digitais e como elas são apresentadas, através de seus laudos.

2.4 Evidencias digitais

Segundo Nelson, Philips e Steuart (2010, p. 150): “evidência digital é qualquer informação armazenada ou transmitida em meio digital”. As evidências digitais estão associadas a dispositivos eletrônicos e periféricos, esses aparelhos são instalados junto ao computador, e estão associados à cena de um crime. Através das evidências digitais é possível coletar e analisar os dados ali inseridos ou contidos em um computador ou dispositivo móvel, discos rígidos, cartão de memória e outros.

Na atualidade, existe um consenso em relação a característica física ou virtual das evidencias digitais. Os tribunais dos EUA, aceitam provas digitais como evidencia física, significa que os elementos digitais, são tratados como objetos tangíveis assim como, documentos em papel ou uma lesão visível, relacionados a incidentes criminais ou civis. Tribunais em outros países ainda estão atualizando suas leis para levar em conta as evidências digitais. Alguns exigem que todas as evidências digitais sejam impressas para serem apresentadas no tribunal.

Para a computação, os insumos de um crime são digitais, visto que toda a informação armazenada dentro desses equipamentos computacionais é composta por bit, sequência lógica representada por zeros e uns. No Brasil, o Código do Processo Penal, determina que quando a infração deixar vestígio, será indispensável o exame de corpo de delito, direto ou indireto, não podendo supri-lo a confissão do acusado.

3. DISPOSITIVOS MÓVEIS E COLETA DE DADOS

A computação móvel e suas aplicações estão cada vez mais presente na vida das pessoas e no cotidiano, modificando a forma como as pessoas realizam suas atividades, quer enviando e-mail, realizado operações bancárias e no acesso as redes sociais (SANCHES et al., 2014).

De acordo com Chainho (2014), os dispositivos móveis transformaram a forma de comunicação entre as pessoas. Tudo isso foi permitido devido ao avanço da tecnologia que pós a capacidade de processamento de um computador e a capacidade de comunicação de um telefone em um único dispositivo. O primeiro dispositivo concebido, para esse proposito foi o PDA[9] *(Personal Digital Assistant).*

[9] PDA é um aparelho concedido para possibilitar a seus usuários acesso à web, e-mail, entre outras funções.

Ainda segundo Chainho (2014), a partir da criação dos PDA's, os avanços da tecnologia tanto em nível de hardware e software, deram origem aos novos dispositivos que são embutidos com ferramentas variadas, tanto para trabalho como para diversão. Essas ferramentas ou aplicações, são a base para o sucesso da análise forense em dispositivos moveis e, é através da compreensão de hardware e software, que se pode analisar as atividades e coletar dados dos usuários e obter provas. Algumas características observadas na maioria dos dispositivos móveis, que podem ser analisadas para a obtenção de provas são: microprocessador, memoria ROM, memória RAM, processador de sinal digital, tela, sistema operacional, GPS, Câmera e dentre outras.

De acordo com Chainho Apud Hoog (2014, p.48), explica que o sistema operacional Android baseado no Linux Kernel 2.6, gerido pela OHA[10] (*Open Handset Alliance*), tornou-se uma plataforma que é atraente para o desenvolvimento, então ele propõe quatro abordagens para investigação forense e coleta de dados no sistema operacional Android, das quais: Analise de Cartão SD; Aquisição Lógico; Aquisição Físico; Chip-off.

Hoog propõem para os especialistas a aquisição lógica como a melhor opção para a análise forense em dispositivos móveis como SO Android, dando como referência a ferramenta de código aberto desenvolvia pela "*viaForensics"*¸ onde se pode obter diversas informações como mostra o quadro 1 abaixo:

[10] OHA é uma aliança de diversas empresas com a intenção de criar padrões abertos para telefonia móvel. Entre as empresas participantes estão Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile e Nvidia, dentre outras.

Quadro 1 — Informações extraída pela aplicação viaForensics

INFORMAÇÕES OBTIDAS
Mensagens de texto (SMS / MMS)
Contatos
Registro de chamadas
Mensagens de e-mail (Gmail, Yahoo, Exchange)
Mensageiro Instantâneo / Chat
Coordenadas GPS
Fotos / Vídeos
Histórico da web
Histórico de busca
Facebook, Twitter e outros clientes de mídia social
Arquivos armazenados no dispositivo
Coleções de música
Compromissos do calendário
Informação financeira
Histórico de compras
Compartilhamento de arquivos

Fonte: Adaptado de Chainho (2014)

Hoog também desenvolveu um método para fazer a extração física dos dados em dispositivos com sistema Operacional Android, no formato YAFFS2 (*Yet Another flash File System*). O YAFFS2 é um sistema de arquivo criado especificamente para dispositivos como memória flash, licenciado sob licença GPL[11] (GNU *public License*), possuindo características importantes como: sistema de arquivos de log estruturado (prevenção contra interrupção de energia), correção de erros, capacidade para lidar com setores defeituosos; rápido e utiliza pouco memória RAM. A desvantagem é que não existem muitas aplicações que deem suporte a este tipo de formato de ficheiros, significando que a análise deve ser feita de forma manual (JUNIOR, 2013, p. 31).

3.1 Melhores prática para coleta de dados

As melhores práticas para obtenção de dados definem métodos que devem levar em conta o procedimento de apreensão do dispositivo móvel, coletas dos dados, exame e processo de documentação, produção de relatórios e confecção de laudos.

[11] GNU General Public License, GNU GPL ou simplesmente GPL, é a designação da licença de software para software idealizada por Richard Matthew Stallman em 1989, no âmbito do projeto GNU da Free Software Foundation.

A etapa de coleta de dados consiste no processo de criação de imagens forenses ou outras etapas para aquisição das informações como: análise de seus periféricos, equipamentos e memórias removíveis e ainda aquisição das informações por meio de navegação manual nas pastas do sistema. Esse sem dúvida é o momento mais técnico onde o conhecimento do perito sobre a plataforma Android e experiência são cruciais e de suma importância (CHAINHO, 2009, p. 14).

De acordo com a ISFS (*Information Security and Forensics*)[12] (2009, p. 14), o objetivo de se desenvolver padrões de melhores práticas é estabelecer métricas que possam servir de referência de qualidade, de princípios de qualidade e abordagens para a detecção, preservação, recuperação, exame e uso de informações digitais para fins forenses.

Ao descrever melhores práticas para a análise forense digital, a ISFS (2019, p. 19) define os fatores chaves e principais responsabilidades que devem ser seguidos em qualquer analise forense pelos profissionais envolvidos, conforme abaixo:

Fatores Chaves: Quatro coisas são fundamentais para todos os exames forenses:

1. A manutenção da integridade dos dados, bem como autenticidade dos dados;
2. Prevenção da contaminação de dados,
3. documentação adequada e abrangente e;
4. Implementação de uma metodologia científica sistemática.

Responsabilidades chave: Todos os profissionais envolvidos em um exame forense devem ter ética e uma profissional responsabilidade de:

1. Mantenha sua objetividade;
2. Apresentar fatos com precisão;
3. Não reter quaisquer descobertas, pois tais ações podem distorcer ou deturpar os fatos;
4. Render opiniões apenas com base no que pode ser razoavelmente demonstrado.

[12] ISFS (Sociedade de Segurança e Informações Forenses), é uma instituição que foi registrada sob a Portaria de Sociedades de Hong Kong em maio de 2000. Tem como missão defender e reforçar o profissionalismo, integridade e inovação em Segurança da Informação e Computação Forense. http://www.isfs.org.hk/about.html.

Além de definir os fatores chaves e as responsabilidades chaves, também oriente que na condução dos exames forense, o especialista em computação deve aplicar, todos os princípios forenses e processuais gerais para lidar com evidências digitais; não realizar quaisquer ações que possam alterar as evidências mantidas, incluindo dados sobre mídia ou computadores; garantir que apenas pessoas qualificadas acessem a evidência digital (ISFS, 2019).

Pode ser necessário, em circunstâncias excepcionais, permitir que outros acessem dados originais em um computador de destino ou outra mídia. Nesses casos, é vital que todas e quaisquer pessoas que acessem os dados são competentes para fazê-lo e são competentes para evidenciar a relevância e as implicações de suas ações (ISFS, 2019).

Documentar todas as atividades relacionadas à apreensão, acesso, armazenamento e transferência de evidências digitais e preservar um registro. Uma terceira parte independente deve poder examinar esses procedimentos documentados de tal forma que este terceiro pudesse repetir o processo, ele alcançaria o mesmo resultado (ISFS, 2019).

O especialista é responsável por todas as ações tomadas com relação a evidência digital. Como responsável, ele deve garantir que esses princípios de boas práticas forense sejam respeitadas. Isso se aplica à posse e acesso a informações contidas em um computador ou dispositivo móvel. É o dever do especialista em Computação Forense encarregado de garantir que qualquer pessoa que acessar provas cumpre estes princípios (ISFS, 2019).

3.2 Busca, apreensão e conservação

É recomendado que antes da extração dos dados do dispositivo móvel, se faça uma correta preservação do mesmo para que ele possa chegar a um perito ou analista pericial na forma mais adequada possível para realização dos exames. Contudo a apreensão tem como principal objetivo preservar as evidências digitais que possam ser uteis e evitar que as provas sejam alteradas.

Uma das questões mais importante é a conservação de forma adequada das evidências que vierem a ser encontradas, a fim de documentá-las posteriormente.

Segundo a ISFS (2009), assim que iniciar a investigação todas as atividades devem ser documentadas, essa documentação deve conter a fonte ou a localização original de todas as evidências, registro das ações tomadas e anotações. Além disso, existe quatro razões citadas pela ISFS que são essenciais pela qual se exige essa documentação, que são: A documentação mostrando evidência em seu estado original, reafirma que são autênticas e inalterado. Como cópias

e originais são difíceis de distinguir, a documentação pode ser necessária para diferenciar entre cópias e originais; em situações em que vários computadores idênticos com configurações idênticas, a documentação é crítica na distinção desses sistemas; e documentação do local original da evidência pode ser útil na reconstrução de um crime.

De acordo com Jansen, Brothers e Ayres (2014, tradução nossa):

> "No processo de busca e apreensão, acena poderá ser fotografada e, se houver a presença de um perito na equipe, em havendo necessidade, será realizado um laudo local onde constarão informações acerca de todas 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".

O perito forense deve documentar todos o processo de exames durante a aquisição das informações e assim que o perito concluir a investigação, os resultados dever passar por algum tipo de revisão, para garantir que os dados foram verificados e que a investigação seja finalizada. A documentação do examinados pode conter informações como a seguinte: a data de começo do exames e tempo de duração, condição física do dispositivo, fotos do dispositivo e componentes individuas, status do telefone, caso encontre-se ligado ou desligado, marcas e modelos, ferramentas utilizadas para aquisição e ferramentas utilizadas para o exame (TAMMA *et al*., 2018).

3.3 Aquisição dos dados

Conforme Jansen, Brothers e Ayres (2014, p. 47) o exame forense começa como a identificação do dispositivo móvel, tipo de dispositivo, sistema operacional e outras características. A avaliação inicial destas características, determinam o caminho a ser seguido na realização de um exame forense e da aquisição da cópia dos dados a serem avaliados. O Tipo de dispositivo móvel e os dados a serem extraídos geralmente determinam quais ferramentas e técnicas devem ser usadas em uma investigação. Desta forma, a aquisição consiste na extração dos dados do dispositivo móvel para posterior análise e deve ser realizada de forma isolada da rede de comunicação móvel [13][14]do aparelho, utilizando-se de hardware e software adequados para obtenção dos dados.

[13] Redes de telefonia móvel é um sistema inteligente de alocação e reuso dos canais através da área de cobertura. Um grupo de canais de rádio é alocado a cada estação base celular para atender uma pequena região geográfica denominada célula.
[14] Para mais informações, visite o site: https://www.gta.ufrj.br/grad/10_1/movel/caracteristicas.html.

Segundo Tamma e Tindall (2015), a fase de aquisição refere-se a extração dos dados do dispositivo avaliado, um obstáculo encontrado nesta fase, está relacionado aos recursos de segurança inerentes ao dispositivo móvel, desta forma a extração dos dados nem sempre e direta e como dito anteriormente, depende do sistema operacional marca, modelo e posteriormente o método de aquisição é decidido.

Conforme a NIST (2014) muitos dispositivos móveis oferecem a capacidade de bloqueio remotamente ou limpeza remota, simplesmente enviando uma mensagem de texto para o dispositivo móvel. Essa característica do dispositivo móvel configura um motivo adicional para desabilitar a conectividade de rede e inclusive dados de entrada, como as mensagens de texto que podem promover modificações no estado atual de armazenamento do dispositivo móvel. Dados de saída, também são indesejáveis, pois a localização atual do GPS pode fornecer a localização geográfica do examinador forense.

Ainda segundo a NIST (2014), existe três métodos básicos para isolar o dispositivo móvel de algum tipo de comunicação que são:

1. colocar o dispositivo no modo avião;
2. desligar o dispositivo e;
3. colocar o dispositivo em um contêiner protegido.

Portanto, antes da extração dos dados o examinador deve evitar que o dispositivo se comunique, com a rede de telefonia ou realize canecões com a rede *WI-FI, bluetooth, IrDA* (infravermelho), para isso pode ser usado equipamentos específico para isolamento da rede, ou ainda pode ser realizado intervenção manual, como pôr o dispositivo em modo avião NIST.

3.4 Ferramentas de extração

É de grande relevância que algumas características sejam observadas nas ferramentas forenses, uma vez que estas ferramentas deverem apresentar dados de tal maneira, que possam ser úteis e manipuladas pelo analisador forense.

De acordo com NIST, as ferramentas de software forense para dispositivos móveis são consideradas diferente das utilizadas em computadores pessoais, e mesmo se diferenciando do ponto de vista de hardware e software suas funcionalidades se tronaram semelhantes. Embora os smartphones possuam em sua grande maioria, sistemas operacionais de código aberto (ou seja, Android), também existe uma grande variedade de smartphone com sistemas operacionais de

código proprietário, que dificultam a interpretação do sistema de arquivos e das estruturas associadas. E ainda, muitos dispositivos móveis com o mesmo sistema operacional pode variar e se diferenciar muito em sua implementação, resultando em uma infinidade de versões de um mesmo sistema, criando assim; um obstáculo para os fabricantes e examinadores de ferramentas forenses para dispositivos móveis. (JANSEN; BROTHERS; AYERS, 2014).

Ainda segundo a NIST, os tipos de software disponíveis para os dispositivos móveis incluem ferramentas comerciais e de código aberto, e ainda ferramentas não forenses que são destinadas ao gerenciamento, teste e diagnóstico do dispositivo e também ferramentas projetadas para adquirir dados de memória interna do aparelho, sem alterar seu conteúdo. Os examinadores forenses fazem uso dessas ferramentas, montando um kit como softwares disponíveis no mercado, operando sobre uma variedade de dispositivos como plataformas distintas, famílias de sistemas operacionais e até mesmo um único tipo de arquitetura de hardware. (JANSEN; BROTHERS; AYERS, 2014).

Portanto devido à grande variedade de dispositivos disponíveis no mercado, modelo, famílias de sistemas operacionais e versões de fabricantes é necessário que tenhamos no mercado ferramentas atualizadas e compatíveis com a realidade e essas ferramentas devem ser validades por uma equipe de examinadores.

3.5 Memorias dos dispositivos móveis

Os dispositivos móveis têm dois tipos principais de memorias voláteis (RAM) e memorias não voláteis (NAND), (Hoog, 2011, p. 125).

A RAM[15] é usada pelo sistema para carregar, executar e manipular partes importantes do sistema operacional, aplicativos ou dados e não é salva na reinicialização. Para a computação Forense, a RAM pode conter informações muito importantes que os aplicativos usam para processar dados, tais como: senhas, chaves de criptografia, nomes de usuários e dados do aplicativo (Hoog, 2011, p. 125).

[15] RAM (Random Access Memory - Memória de Acesso Aleatório) é um hardware de armazenamento randômico e volátil de memória. Isto significa que esta peça armazena dados de programas em execução enquanto o dispositivo está ligado.

Ao contrário da RAM, o flash NAND[16] não é volátil e, portanto, os dados são preservados mesmo quando o dispositivo está sem energia ou reinicializado. O flash NAND é usado para armazenar não apenas arquivos do sistema, mas também partes significativas dos dados do usuário. (Hoog, 2011, p. 127).

Os smartphones são dispositivos dotado de memória flash, que é uma memória do tipo não volátil e que é usada principalmente em cartões de memória, e unidades flash USB, para armazenamento de estado solido. Para a computação forense é importante estudar esse tipo de memória, uma vez que elas possibilitam a aquisição física de dados, que é uma cópia bit a bit do armazenamento físico do sistema de arquivo inteiro, possibilitando a aquisição de informações pelo acesso direto à memória flash do dispositivo (TOMMA et al., 2018).

3.6 Código de segurança e controle de acesso em dispositivos móveis

Os dispositivos móveis possuem muitas formas de controle de acesso a suas aplicações, para possibilitar maior segurança a seus usuários.

Conforme Jansen, Brothers e Ayres (2014), um mecanismo de controle de segurança em dispositivos móveis é o cartão SIM. Com o SIM bloqueado o dispositivo só poderá ser acessado através da digitação do código PIN configurado pelo usuário. Caso este código seja digitado incorretamente por três vezes, o cartão solicitará o código PUK. O PUK vem pré-configurado de fábrica e é fornecido no momento da aquisição do cartão SIM. Outra forma de obter o PUK é por meio de informação da operadora de telefonia ao qual o cartão está vinculado. Caso o PUK seja digitado errado por 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.

3.7 Memórias removíveis

As memórias removíveis na atualidade são muito utilizadas em celulares, pois sua capacidade de armazenamento de dados de um dispositivo móvel é grande.

[16] NAND Flash é um tipo de memória flash. A memória flash é a memória que é "não- volátil", o que significa que ao contrário da memória de acesso aleatório (RAM), memória flash mantém os dados mesmo que não haja uma perda de energia. NAND Flash é uma forma amplamente utilizada de memória que oferece alta velocidade de acesso e grande armazenamento de dados em comparação com outros tipos de memória flash.

Conforme a Jansen, Brothers e Ayres (2014), dispositivos móveis podem fornecer ao usuário uma interface para um cartão de memória. As ferramentas forenses de dispositivos móveis que adquirem o conteúdo de um cartão de memória, normalmente realizam uma aquisição lógica. Se o dispositivo for encontrado em um estado ativo, a memória interna do dispositivo móvel deve ser adquirida antes de remover e executar uma aquisição física da mídia associada, como por exemplo, cartão microSD). Caso contrário, se o dispositivo for encontrado em um estado desligado, uma aquisição física da mídia removível deve ser realizada antes da aquisição da memória interna do dispositivo móvel. Como em qualquer tipo de aquisição, a ferramenta forense pode ou não ter a capacidade de decodificar os dados recuperados armazenados no cartão.

4. BREVE INTRODUÇÃO SOBRE O SISTEMA OPERACIONAL ANDROID

Em 2005, o Google começou a investir dinheiro em empresas que achavam que seriam lucrativas e dessa maneira resolveu adquirir a empresa Android Inc., que foi fundada em 2003, por Andy Rubin, Rich Miner, Nick Sears e Chris White. Essa empresa virá a se tornar um dos melhores negócios de todos os tempos (TAMMA et al., 2018).

Após dois anos e sem notícias operando sobe sigilo, a empresa descreveu-se como desenvolvedora de software para telefones móveis e que revolucionaria a forma como os aparelhos celulares operaram. Então, Rubin ficou com o Google para ser o pioneiro do Android como sistema operacional. Com a aquisição da empresa Android Inc. ficou claro que o Google tinha interesse no mercado de telefones móveis, portanto, Rubin desenvolveu juntamente com a sua equipe um sistema operacional poderoso e flexível, construído baseado em um Kernel Linux. (TOMMA et al., 2018).

Por volta de 2007, a *Open Handset Alliance* (OHA), que configura um grupo de empresas de tecnologia, fabricantes de dispositivos, chipsets e operadoras sem fio, foi fundada com o propósito de criar padrões abertos para a plataforma Android, juntos eles desenvolveram o Android, a primeira plataforma móvel aberta e gratuita, baseada no Kernel Linux 2.6. Posteriormente, foi lançado o HTC Dream, primeiro dispositivo a rodar o sistema operacional Android, após isso o Android vem crescendo exponencialmente e sendo de longe o sistema operacional mais usado em todo o mundo (TAMMA et al., 2018).

4.1 Visão geral de versões do Android

A várias versões lançadas do Android e cada uma traz suas vantagens e desvantagem que foram determinantes para a evolução do sistema. É interessante ressaltar que as versões 1.0 (*Apple pie*) e 1.1 (*Banana Bread*), seriam versões *Alpha* e *Betas*, respectivamente.

O quadro 2 abaixo, monstra as versões do Android e uma breve descrição de suas funcionalidades (BARBOSA, 2017, p. 5).

Quadro 2— Versões do sistema operacional Android

(Continua)

VERSÃO	NOME DA VERSÃO	RELEASE	FUNCIONALIDADES
Android 1.0	Apple pie	2008	Versão Alpha, conhecida como a primeira versão comercial do Android, lançada em setembro de 2008, junto com a chegada do primeiro dispositivo equipado para receber o sistema operacional Android, o HTC Dream. Os recursos incorporados a esta versão foram: aplicação Android Market que realiza downloads e atualiza aplicativos através do aplicativo Market; Web Browser para exibir, dar zoom e suporte a páginas em HTML e XHTML; notificações na barra de status; suporte à câmera, contudo nesta versão não havia opções de modificar a resolução da câmera, qualidade, balanço de branco, entre outros.
Android 1.1	Banana Bread	2009	Versão Beta, foi lançada em fevereiro de 2009, primeira atualização comercial do sistema Android. Os principais recursos adicionados foram: suporte para letreiros em layouts; tempo de limite de tela maior para chamadas padrão feitas pelo viva-voz; capacidade de salvar anexos de mensagens; comentários e mais detalhes sobre diversos assuntos pesquisados no Google Maps.
Android 1.5	Copcake	2009	Lançado em abril de 2009 baseado no Kernel Linux 2.6.27, produzindo várias melhorias para o sistema operacional como na parte do GPS, câmera, upload de vídeos e fotos para o YouTube, entre outros. Porém, a principal novidade foi o lançamento do primeiro Android com o touch screen e o teclado virtual. Também surgiu a ideia de widgets - pequenos aplicativos que são executados na tela inicial.

Quadro 2— Versões do sistema operacional Android

(Continuação)

VERSÃO	NOME DA VERSÃO	RELEASE	FUNCIONALIDADES
Android 1.6	Donut	2009	Foi lançado em setembro de 2009 baseado no kernel Linux 2.6.29. Nesta versão, o sistema fala e escuta através da conversão de texto em voz (TTS) e conversão de voz em texto (STT). Com o auxílio das pesquisas de voz, a home do Android garantiu mais funcionalidades, facilitando o uso do sistema operacional pelo usuário. Além da API de programação para uso de text-to-speech, os principais recursos desta versão foram: interfaces para a programação de aplicativos com reconhecimento de gestos; suporte para resoluções de tela (320×240 e 800×480); capacidade de selecionar vários recursos simultaneamente.
Android 2.0 e 2.1	Eclair	2009	Foi lançado em outubro de 2009 e posteriormente atualizado em janeiro de 2010. Trouxe diversas melhorias ao sistema, destacando a parte das câmeras e mapas. Nessa versão foi projetado o suporte ao HTML5 e a múltiplas contas do Google e sincronização.
Android 2.2	Froyo	**2010**	Foi lançado em maio de 2010 baseado no kernel Linux 2.6.32, trazendo melhorias de desempenho para o sistema operacional, como o compilador JIT (Just in Time). Nessa versão foram acrescentados os seguintes recursos: Wi-Fi Hotspot (compartilhamento de internet com computadores com hotspots para até 8 dispositivos) e USB Tethering; suporte ao flash; introdução da barra fixa de atalhos: navegador, telefone e launcher; Android Market pode atualizar aplicativos automaticamente. Além de que os desenvolvedores puderam criar apps que aprimoram a segurança do aparelho, como as telas de bloqueio e a possibilidade de armazenar aplicativos no cartão SD.
Android 2.3	GingerBread	2010	Foi lançado em dezembro de 2010 baseado no kernel Linux 2.6.35, trazendo novidades na câmera, possibilitando alternar entre a câmera frontal e traseira. Além disso, apresentou melhorias na funcionalidade de copy-paste, excelente ganho com o gerenciamento de bateria, suporte à NFC (Near Field Comunicattions) e aos sensores de movimento.

Quadro 2 — Versões do sistema operacional Android

(Continuação)

VERSÃO	NOME DA VERSÃO	RELEASE	FUNCIONALIDADES
Android 3.0	HoneyComb	2011	Foi lançado em fevereiro de 2011 baseado no kernel Linux 2.6.36, com o sistema operacional focado nos tablets e aprimorando a experiência dos usuários. Entre as novidades desta versão, se evidenciam: barra de ação, permitindo acessar navegação, widgets e outros tipos de conteúdo; interface remodelada e novo sistema multitarefas.
Android 4.0	Ice Cream Sandwich	2011	Foi lançado em outubro de 2011 baseado no kernel Linux 3.0.1. Nesta versão, várias novidades foram adicionadas tanto no funcionamento quanto no design. Um dos objetivos desta versão do Android era unificar as interfaces separadas do Android para smartphones (Gingerbread) e para tablets (Honeycomb), o que foi realizado com sucesso. Além disso, a empresa aprimorou as "diretrizes para interface humana" do Android, ou seja, simplificou e modernizou a experiência global com o sistema operacional. Adicionou, ainda, novos recursos ao Android Ice Cream Sandwich, como a nova tela inicial e o suporte à tecnologia NFC.
Android 4.1	Jelly Bean	2012	Foi lançado em junho de 2012 baseado no kernel Linux 3.0.31. Nesta versão, houve ganhos com relação ao desempenho do sistema, trazendo uma interface renovada e notificações expansíveis. Além disso, incluiu a tecnologia Photo Sphere, para produção de imagens em 360º e a possibilidade de realizar gestos na tela de bloqueio para acessar rapidamente a câmera do celular.
Android 4.4	Kik Kat	2013	Foi lançado em outubro de 2013, elevando o desempenho do sistema, otimizando a memória e aperfeiçoando a tecnologia touchscreen a respostas mais rápidas e com maior precisão. Além disso, trouxe aperfeiçoamentos no Bluetooth, Print Framework, sensores, NFC. Nesta versão, também foi criada a API de Transitions para animações personalizadas.

Quadro 2 — Versões do sistema operacional Android

(Conclusão)

VERSÃO	NOME DA VERSÃO	RELEASE	FUNCIONALIDADES
Android 5.0	LolliPop	2014	Foi lançado em novembro de 2014. O Lollipop trouxe várias novidades e uma nova política visual, denominada de Material Design, na qual passou a guiar não apenas o Android, mas todos os produtos da empresa, inclusive os serviços web. Além disso, houve melhorias nas notificações, que aparecem também na tela de bloqueio (Lock Screen) e as head-up notificações, que aparecem no topo da tela com alta prioridade. Outra novidade desta versão foi o projeto Volta, que trouxe ferramentas para auxiliar a análise do uso da bateria nos apps.
Android 6.0	Marshmallow	2015	foi lançado em outubro de 2015. Dentre as principais novidades desta versão, pode-se citar a inclusão de um novo modelo de permissão no qual apenas oito categorias precisam ser ativadas na primeira vez em que o usuário usa o aplicativo. Sendo assim, elas deixam de ser concedidas automaticamente no momento em que o usuário faz o download. Além disso, oferece suporte nativo para o reconhecimento de impressões digitais e também apresenta um novo sistema de gestão de energia, chamado “Doze.
Android 7.0	Nougat	2016	Foi lançado em agosto de 2016. As principais características desta versão são: encriptação nativa no qual smartphones encriptados vão funcionar mesmo após reiniciação inesperada; API Job Scheduler tornando o smartphone mais rápido; uso da tecnologia Vulkan.
Android 8.0	Oreo	2017	Foi lançado na Google I/O em agosto de 2017. Apresenta as seguintes novidades: facilidade em acompanhar o consumo de bateria por parte dos aplicativos e também pelos próprios serviços do Android; o recurso de conversão de texto em fala, adquiriu melhorias, como o suporte a outros idiomas além do nativo do sistema.

Fonte 2 — Adaptado de Barbosa (2017)

4.2 A Arquitetura do sistema operacional android

Segundo Hoog (2011, p. 96, tradução nossa), o sistema operacional Android possui uma arquitetura de alto nível especialmente para procedimentos de segurança. Baseado no Kernel do Linux 2.6, fornece o software fundamental para iniciar e gerenciar tanto o hardware quanto os aplicativos Android, além disso, seu Kernel fornece funcionalidades bastantes extensa das quais, nós podemos visualizar na figura 2 abaixo.

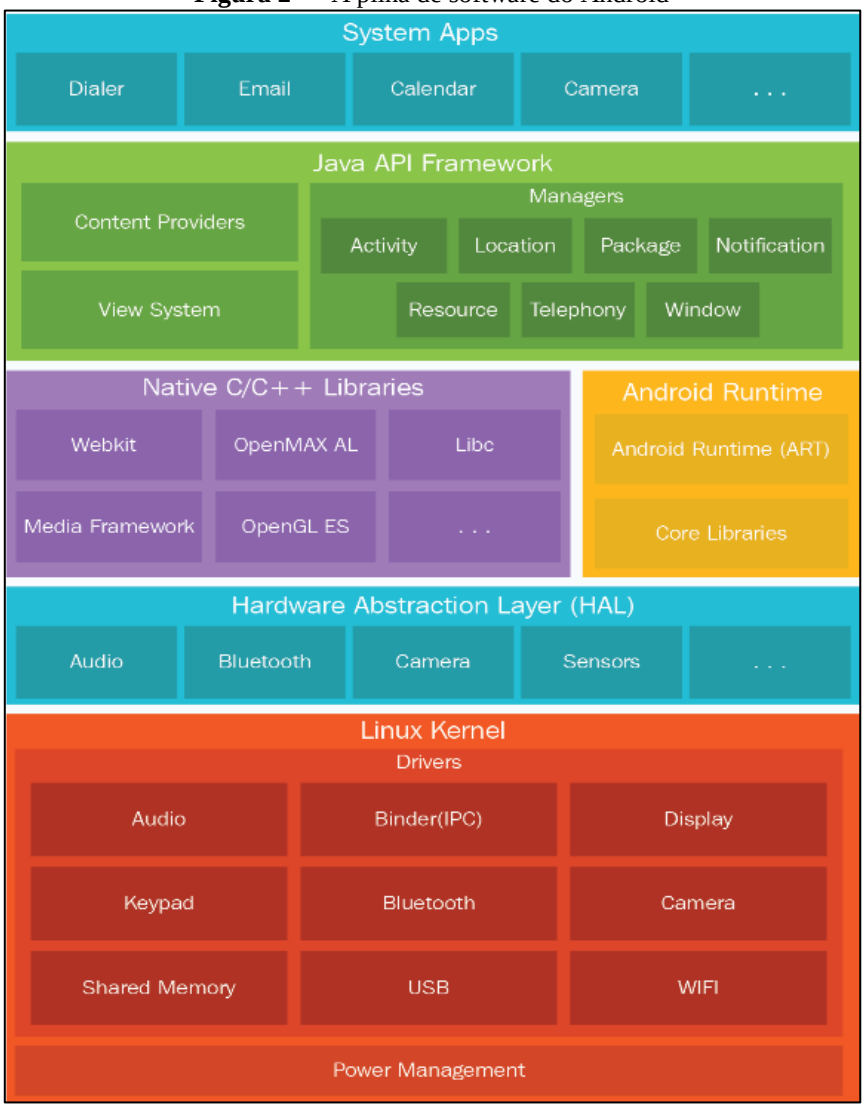
As funções consideradas de nível mais baixo, incluem gerenciamento de energia, *WIFI, USB, Shared Memorie, Camera, Bluettoth, Keypad, Display, Binder (*IPC*)* e *Audio*. Todas esses *Drivers* são gerenciados pelo *Linux Kernel*. Sob uma perspectiva forense, o *driver* mais importante é *Shared Memory* (HOOG, 2011).

Após o Kernel um conjunto de bibliotecas está disponível, fornecendo funcionalidade importante, tanto para os desenvolvedores quanto para os proprietários dos dispositivos. Estes incluem as bibliotecas *WebKit* para renderização HTML no navegados e aplicativos de outros desenvolvedores. Outras bibliotecas, lidam com fontes, display, diversas mídias e comunicações seguras usando o SSLs[17] (*Secure Socket Layers*), (HOOG, 2011).

As bibliotecas principais são empacotadas com uma máquina virtual (VM) Java, que é o local onde os aplicativos são executados. Por fim, o SDK fornece permissão a esses recursos através de APIs e uma estrutura de aplicativos (HOOG, 2011).

[17] SSL é uma ferramenta de encriptação de páginas antes de serem transmitidas pela internet que autentifica as partes envolvidas. É muito utilizada para pagamentos online com cartão de crédito.

Figura 2 — A pilha de software do Android



Fonte: Arquitetura do Android.
https://developer.android.com/guide/platform/

4.3 A Máquina virtual dalvik

A Máquina Virtual *Dalvk* (Dalvik VM), foi desenvolvida pelo Google para proporcionar um ambiente de aplicativos eficiente e seguro. Com o objetivo de proporcionar segurança, cada aplicativo é executado em sua própria *VM Dalvik*. Desta forma, a *VM Dalvik* foi desenvolvida para que muitas VMs (Máquinas Virtuais), pudessem rodar de uma só vez em um único dispositivo Android.

A VM Dalvik depende do Sistemas Operacional Linux para fornecer funções de baixo nível, para poder ter acesso as bibliotecas e hardware, gerenciamento de ameaças de segurança, gerenciamento de memorial (HOOG, 2011).

Ainda segundo Hoog (2011), os aplicativos executados em uma VM Dalvik possuem um formato especial, chamado de arquivo Executável *Dalvik* (.dex), que possibilita que os desenvolvedores possam escrever e compilar seus programas com o JDK (Java Development Kit).

4.3.1 O Formato dex.

Segundo (TAMMA et al., 2018, tradução nossa), na plataforma Android, todos os aplicativos instalados no dispositivo Android, serão gravados na linguagem de programação Java.

Quando um programa Java é compilado, a resultante desta compilação é um arquivo chamado de *bytecode*[18]. Neste contexto, uma máquina virtual é uma aplicação que possibilita a execução de outras aplicações em plataformas diferentes.

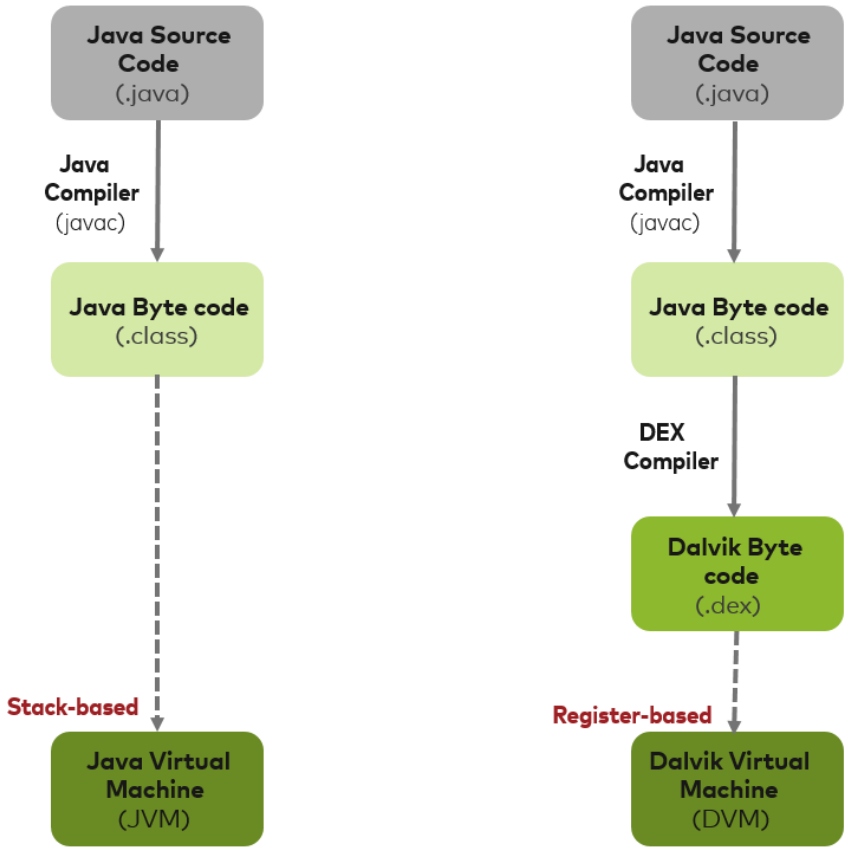
A JVM é uma máquina virtual que pode executar os *bytecode* mencionados anteriormente, porém no Android esses *bytecodes* serão executados pela DVM (Máquina Virtual Dalvik). O DVM executa o *bytecode* Dalvik, que é um *bytecode* Java convertido pelo compilador "Dex". Dessa forma, os arquivos ".class" são convertidos em arquivos ".dex", usando a "ferramenta dx".

O *bytecode* da Dalvik, quando comparado com o *bytecode* Java, é mais adequado para ambientes com pouca memória e baixo processamento (TAMMA et al., 2018, tradução nossa).

A figura 3 a seguir fornece uma visão de como o DVM do Android difere da JVM do Java.

[18] Bytecode é o resultado de um processo semelhante ao dos compiladores de código-fonte que não é imediatamente executável. O bytecode é interpretado numa máquina virtual, que fará a execução, dessa forma o bytecode é um estágio intermediário entre o código-fonte (escrito numa linguagem de programação específica), e a aplicação final, possibilitando portabilidade à aplicação.

Figura 3 — Compilação de Bytecode JVM e DVM



Fonte: Comparação entre JVM e DVM
http://androidjavapoint.blogspot.com/2017/04/closer-look-at-android-runtime-dvm-vs.html

É importante notar que o *bytecode* da JVM é composto por um ou mais arquivos ".class", dependendo do número de arquivos Java que compõem o aplicativo. O *bytecode* da Dalvik é composto por apenas um arquivo ".dex", e cada aplicativo Android executa sua própria instância da DVM (Dalvik Virtual Machine).

A JVM tradicional e a Dalvik VM [19]diferem em muitos aspectos, principalmente no que tange à portabilidade e suporte a bibliotecas de terceiros. Outros recursos como gerenciamento de memória e compilação JIT (Just-in-Time) também foram profundamente alterados, tornando-a mais econômica em termos de consumo de memória e mais performática quanto ao uso de CPU.

[19] Para mais informações visite o site: http://www.luiztools.com.br/post/tudo-sobre-maquina-virtual-dalvik-do-android/

4.4 Android Software Development Kit

O *Android Software Development Kit* (SDK) consistem em um conjunto de ferramentas utilizadas no desenvolvimento de aplicações para a plataforma Android. A cada versão lançada, um conjunto de *API Level* são disponibilizadas pelos desenvolvedores.

O API *Level* (ou nível da API) é um valor inteiro que identifica unicamente a revisão da API da estrutura de trabalho oferecida pela versão da plataforma Android, que pode ser utilizada pelos aplicativos para interagir com o sistema Android subjacente. A Quadro 3 abaixo, especifica o nível de API permitido em cada versão da plataforma Android.

Quadro 3 — Versões do Android e Nível de API correspondente.

VERSÃO DA PLATAFORMA	NÍVEL DA API	VERSION_CODE
Android 7.0	24	N
Android 6.0	23	M
Android 5.1	22	LOLLIPOP_MR1
Android 5.0	21	LOLLIPOP
Android 4.4w	20	KITKAT_WATCH
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
VERSÃO DA PLATAFORMA	NÍVEL DA API	VERSION_CODE
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1

Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Fonte: Plataforma Android e nível de API correspondente
https://developer.android.com/guide/topics/manifest/uses-sdk-element?hl=pt-br#ApiLevels

Segundo Hoog (2011):

> "O kit de desenvolvimento de software (SDK) do Android fornece não apenas as ferramentas para criar aplicativos que são executados na plataforma Android, mas também fornece documentação e utilitários que podem ajudar significativamente na análise forense ou de segurança de um dispositivo".

Deve-se frisar, que um entendimento completo do Android SDK fornecerá muitos insights sobre os dados e o dispositivo, além de utilitários importantes que podem ser usados em investigações forenses.

Segundo Tamma et al. (2018):

> Durante uma investigação forense, o SDK ajuda a conectar e acessar os dados no dispositivo Android. O Android SDK é atualizado com muita frequência, por isso, é importante verificar se sua estação de trabalho também está atualizada. O Android SDK pode ser executado no Windows, Linux e OS X.

Conforme citado acima é importante compreender o conjunto de ferramentas disponibilizadas pela plataforma SDK e a importância que seus utilitários têm para a análise forense e segurança dos dispositivos, além de proporcionar conexão e acesso aos dados do dispositivo.

4.4.1 Android NDK

O *Native Developmente Kit* (NDK) [20] é um conjunto de ferramentais que permite usar código C e C++ em aplicações para a plataforma Android. É possível usá-lo para compilar a partir de seu próprio código fonte ou aproveitar as bibliotecas pré-compiladas, possibilitando um desempenho melhor. Por outro lado, o NDK não compensa a complexidade adicional que inevitavelmente insere no processo de desenvolvimento. Mas, é útil em casos em que precise:

[20] Para mais informações visite o site: https://devloper.android.com/ndk/guides/

ganho de desempenho para aplicativos como jogos ou simulação física; reutilização de bibliotecas C ou C++, dentre outras.

Segundo HOOG 2011, quando um desenvolvedor escreve o código em uma plataforma (por exemplo MAC X), mas compila para uma CPU específica, a técnica é chamada de compilação cruzada de um aplicativo. O NDK facilitará bastante esse processo de compilação, uma vez que fornecesse um conjunto de bibliotecas que o desenvolvedor poderá usar.

Ainda segundo Hoog (2011, tradução nossa):

> Do ponto de vista forense e de segurança, a compilação cruzada é um importante componente para pesquisa e desenvolvimento de novas técnicas e explorações. Enquanto A maioria dos analistas forenses e engenheiros de segurança não precisa compilar o código, entender como o processo funciona e qual o papel que desempenha no processo é importante.

De acordo com o Hoog (2011), do ponto de vista da forense e da segurança, a ferramenta de *cross-compiling* (compilação cruzada), ou seja; compilar um código em uma plataforma tendo como alvo outra) é um importante ponto de estudo, uma vez que seus componentes podem ser reutilizados.

4.4.2 Android virtual devices

Junto com o SDK do Android existe um emulador de plataforma que possibilita a criação de máquinas virtuais para diferentes versões do Android. Essa ferramenta é conhecida como *Android Virtual Device* (AVD) [21]e permite a definição de características de um celular com o sistema operacional Android. Antes de criar as *VM Android* para cada versão do sistema operacional é preciso fazer o download de cada versão da API. Isso é feito no *SDK Manager* que faz parte do SDK. Nele são assinaladas as versões do Android com as quais se deseja trabalhar.

A criação de máquinas virtuais no SDK Manager tem grande importância para os desenvolvedores que necessitam validar suas aplicações com versões específicas do sistema, mas também tem grande valor para a investigação forense.

Conforme HOOG 2011:

> O emulador é especialmente útil para desenvolvedores para criar aplicativos personalizados. No entanto, há um grande valor para o analista forense e engenheiro de segurança, pois você pode criar um perfil de como os aplicativos são executados em um

[21] Para mais informações visite o site: https://developer.android.com/studio/run/managing-avds?hl=pt-br.

> dispositivo. Isso pode ser importante para validar suas descobertas em uma investigação ou para testar como uma ferramenta forense afeta um dispositivo Android.

E conforme Tomma et al. (2018), um dispositivo virtual Android ou AVD tem muita importância sobre o ponto de vista forense. Esses Emuladores são úteis para tentar entender como um aplicativo se comporta e executam em um dispositivo, para confirmar certas descobertas em uma investigação forense. Além disso, disso antes de instalar uma ferramenta forense em um dispositivo real, o Emulador AVD pode ser usado para testar como uma ferramenta funciona e altera o conteúdo em um dispositivo Android.

4.4.3 Android debug bridge – ADB Tool

O Android Debug Bridge (ADB) é considerado um dos componentes mais importantes na análise forense do Sistema Operacional Android. O ADB é uma ferramenta de linha de comando que permite a comunicação com o Android e controla-lo via linha de comando (TOMMA et al., 2018).

Segundo Hoog (2011), o ADB é composto de três componentes:

1 - Cliente: serviço ADB cliente em execução na estação que tentará a comunicação com o Android.

2 - Servidor: serviço ADBD (*Android Debug Bridge Daemon*) em execução também na estação que tentará a comunicação com o Android.

3 - Serviço (*daemon*): serviço ADBD em execução no dispositivo Android

O serviço ADBD precisa estar em execução no dispositivo Android.

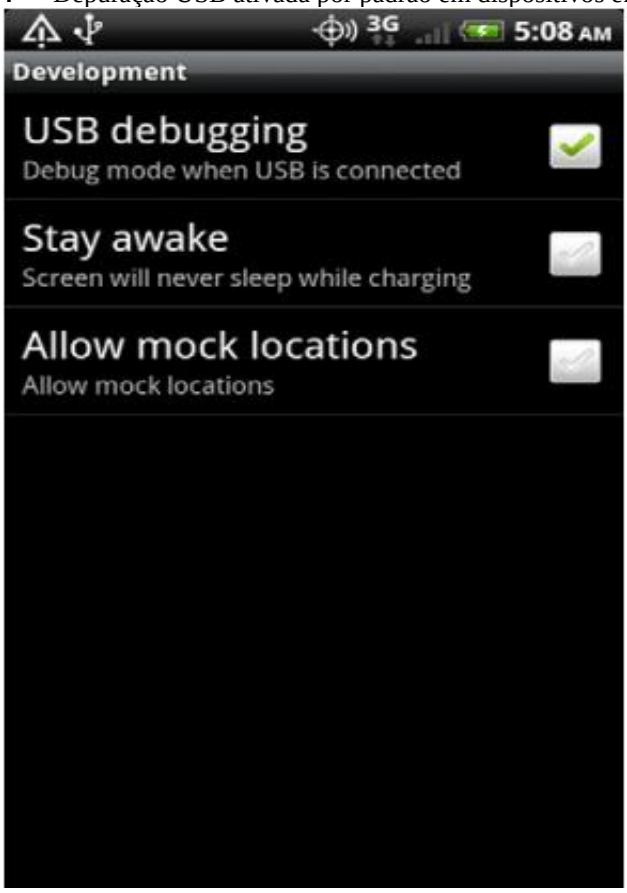
4.4.3.1 USB debugging (Depuração USB).

Como explica Roog (2011), uma interface USB é importante para expor o *Android Debug Bridge* (ADB), permitindo aos Analistas Forenses se comunicarem e controlarem o dispositivo Android via USB.

Para que a comunicação ocorra com os dispositivos “não emulados”, devemos habilitar explicitamente a depuração USB em suas configurações, no entanto; um AVD (em execução no

emulador), terá a depuração via USB ativada por padrão (ROOG, 2011). A figura 4 abaixo mostra a configuração padrão do ADB em dispositivos emulados.

Figura 4 — Depuração USB ativada por padrão em dispositivos emulados

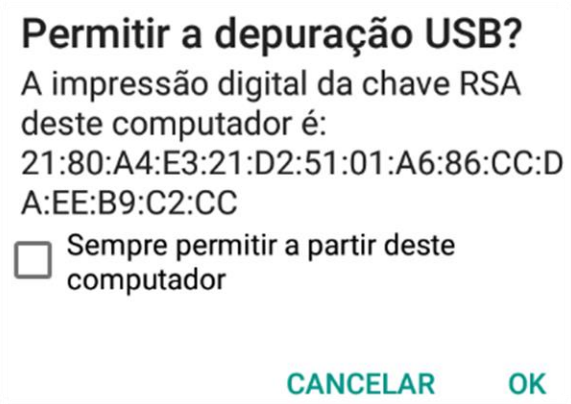


Fonte: Elaborado pelo próprio autor.

Uma vez que a Depuração USB estiver ativada, o dispositivo executara o *daemon adb* (ADBD) em segundo plano e aguardará uma conexão via USB. O *daemon* será executado sob a conta *shell* do usuário sem privilégios, limitando o acesso aos dados. AVD e dispositivos físicos que possuem acesso root habilitado irá executar o adbd como root tendo acesso ilimitado sobre os sistemas (ROOG, 2011).

Em versões mais recentes do Android, sempre que um dispositivo estiver com a Depuração UBS ativa e conectada via UBS, ela exibirá um aviso de segurança, como mostra a figura 5 abaixo.

Figura 5 — Mensagem solicitando permissão de acesso via USB



Fonte: Elaborado pelo próprio autor.

4.5 Sistemas de arquivos

A computação forense está preocupada primeiramente, com os dados que podem ser recuperados em dispositivos móveis para investigação. Como a plataforma Android é baseada no Linux, podemos considerar os artefatos encontrados nesse ambiente e os artefatos novos, tais como: VM *Dalvik* e os sistemas de arquivos YAFFS2 e ainda, a grande complexidade dos sistemas de arquivos distribuídos por diferentes fabricantes (ROOG, 2011).

Ainda conforme Roog (2011), os tópicos que abordam sistemas de arquivos, arquivos dos sistemas *Android* e outros artefatos do sistema é um assunto extremamente densos e complexo, contudo, essas características são o centro do que os analistas forenses devem entender sobre os dispositivos *Android*.

Roog (2011) cita que a variedade de informações que são armazenadas pelo usuário ou pelos aplicativos instalado nos sistemas são enormes. Os aplicativos são a principal fonte desses dados e existe um número de fontes para esses app, que podem ser: aplicativos que acompanham o Android; aplicativos instalados pelo fabricante; aplicativos instalados pela operadora sem fio; aplicativos adicionais do *Google/Android* e aplicativos instalados pelo usuário, normalmente no *Android Market*.

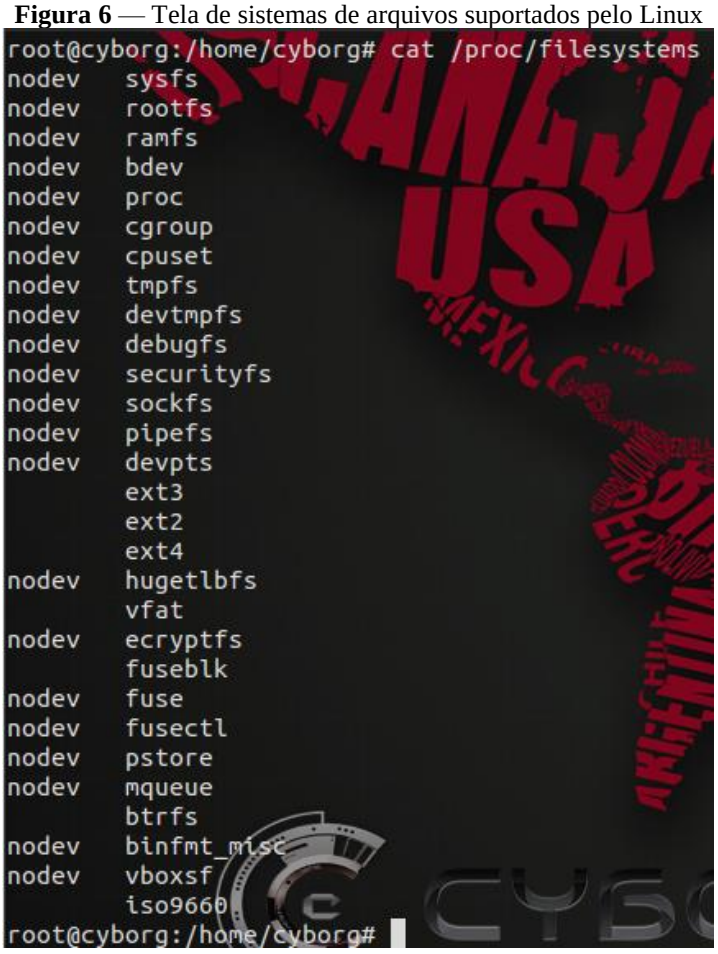
Uma amostra das fontes de dados que podemos ser encontradas são: Mensagens de texto (SMS/MMS), Contatos, Registro de chamadas, Mensagens de e-mail (Gmail, Yahoo, Exchange), Mensageiro Instantâneo/ Chat, Coordenadas GPS, Fotos / Vídeos, Histórico da web, Histórico de busca, Facebook, Twitter e outros clientes de mídia social, Arquivos armazenados no

dispositivo, Coleções de música, Compromissos do calendário, Informação financeira, Histórico de compras, Compartilhamento de arquivos (ROOG, 2011).

Os sistemas de arquivos suportados pelo Sistema Operacional Android podem ser visualizados, verificando o conteúdo do arquivo *filesystems* na pasta "proc". O conteúdo deste arquivo pode ser visualizado usando o seguinte comando:

```
roo@yborg:/home/cyborg cat /roc/filesysems
```

Figura 6 — Tela de sistemas de arquivos suportados pelo Linux



Fonte: Elaborado pelo próprio autor.

4.5.1 Fat32

O sistema de arquivos FAT32, concebido pela Microsoft é reconhecido pela maioria dos dispositivos que utilizam a plataforma *Android* e também é suportado por quase todos os sistemas operacionais, incluindo *Windows*, *Linux* e *macOS*, possibilitando que esses sistemas

operacionais possam ler, modificar e excluir facilmente os arquivos presentes na partição FAT32 dos dispositivos que utilizam o *SO Android*. (TOMMA et al., 2018).

A razão pela qual os sistemas de arquivo das partições utilizadas pelo usuário ser diferente das partições utilizadas pelo *SO Android* é a compatibilidade com outros sistemas operacionais. Como dito anteriormente, sistema como *Linux*, *Windows* e *macOS*, trabalham bem com esses sistemas de arquivos. No *Linux*, partições FAT32 são denominadas de VFAT, que é uma extensão dos sistemas de arquivos FAT16 e FAT32. No SO *Android* existem três lugares que utilizam esse sistema de arquivos, que são:

/mnt/sdcard

/mnt/secure/asec

/mnt/emmc

4.5.2 YAFFS2

O formato YAFFS2[22] (*Yet Another Flash File System*) é um sistema de arquivo *open source single-threaded*, lançado em 2002. Foi projetado para ser rápido quando utilizado com memórias do tipo *flash NAND*. O YAFFS2 utilizada banda OOB, que muitas vezes não é capturada ou decodificada durante uma aquisição forense, o que dificulta a análise (TOMMA et al., 2018).

Sendo um sistema desenvolvido para memória do tipo *flash NAND*, o YAFFS2 atende algumas características importantes, que citadas por Roog (2011):

a) um sistema de arquivos estruturado em log (que protege os dados mesmo por meio de energia inesperada interrupções);

b) correção de erros;

c) capaz de lidar com blocos ruins;

d) rápido e utiliza pouca memória RAM.

[22] O Yaffs2 foi projetado e escrito por Charles Manning, da Whitecliffs, Nova Zelândia, para a empresa Aleph One. Yaffs1 foi a primeira versão deste sistema de arquivos e foi projetada para os chips NAND atuais com tamanho de página de 512 bytes (+ 16 byte de reposição (OOB; Out-Of-Band)).

Uma particularidade das memórias do tipo *NAND flash* é que, quando um arquivo é atualizado, sua versão anterior não é imediatamente apagada porque a memória não permite reescrita (primeiro é preciso apagar todos os blocos de memória – como num processo de preparação), assim, enquanto não for executado o *garbage collector*, o arquivo permanecerá num diretório oculto e será possível recuperá-lo, apenas não será apresentado pelo sistema. Na atualidade, o YAFFS2 não é suportado por versões mais recentes do kernel, mas alguns fabricantes de dispositivos móveis ainda podem continuar a suportá-lo, (ROOG, 2011).

4.5.3 Partições do sistema de arquivos

O sistema de arquivos do *Android* é dividido em três partições principais:

a) A partição do sistema /sistema: Esta partição contém arquivos relacionados ao sistema diferentes do *kernel* e do disco RAM. Essa pasta nunca deve ser excluída, pois isso tornará o dispositivo não incivilizável. O conteúdo desta partição pode ser visualizado usando o seguinte comando: # `cd /system;`

b) A partição `/sdcard/`: que é uma partição livre, onde todos os aplicativos podem gravar dados e ler a partir dela, bastando ter a permissão *WRITE_TO_EXTERNAL_STORANGE*;

c) A partição de dados do usuário `/data/`: onde podem estar aplicações e diversos tipos de mídia. Para um perito, um dos diretórios mais importantes desta partição é o `/data/data/`, onde ficam as bases de dados das aplicações.

4.6 Logs das aplicações do sistema

Os arquivos de log guardam informações importantes sobre as atividades das aplicações quando elas são executadas, desta forma é possível extrair informações importante destes arquivos.

Segundo HOOG 2011, estes sistemas de arquivos padrão do Linux, configuram como uma área adicional onde os analistas forenses podem localizar arquivos e informações relevantes

para uma investigação. Os arquivos de log e depuração fornecem uma visão das informações dos aplicativos e dos sistemas que os executa, desta maneira é possível extrair informações simplesmente examinando os arquivos de log e depuração.

4.6.1 Log do Kernel do Linux

O Kernel do Sistema Operacional Linux controla todo o hardware de um computador, alocando memória se necessário e executando uma aplicação caso seja requerido.

Conforme HOOG 2011, o kernel do Linux é uma interface de baixo nível do sistema operacional Linux, que fornece acesso ao hardware de um dispositivo como smartphones. O papel do Kernel do Linux é geral e para todas as funcionalidades dos dispositivos. Ele tem capacidade de registar eventos e atividades-chaves e, é acessível em um dispositivo Linux e portanto, pelo Android.

O Kernel do Linux possui quatro atividades primárias das quais são: gerenciar a memória do sistema, gerenciar as aplicações do sistema, gerenciar o hardware do dispositivo e gerenciar o sistema de arquivos. Como dito anteriormente, sendo o kernel o núcleo de todas as atividades do dispositivo com sistemas operacional Android, sua capacidade de registrar eventos chaves é altamente desenvolvida.

4.6.2 Log dos Apps

Além dos logs gerados pelo Kernel do Linux e pelo próprio Android, os desenvolvedores de aplicativos para o sistema Android, habituam-se adicionar as suas aplicações a capacidade de gerar os seus próprios arquivos de logs.

No Android, toda a interação do usuário com o sistema é feita através de uma aplicação. Algumas aplicações são instaladas pelo usuário e outras são disponibilizadas pelo fabricante. Por exemplo, as funções rotineiras como utilização dos contatos, chamadas, SMS, dentre outras; são executas por meio de seus respectivos aplicativos. Assim, a análise de aplicativos para Android é crucial durante o curso de uma investigação (TAMMA et al., 2018).

Vários aplicativos de terceiros, como o *WhatsApp*, o *Facebook*, o *Skype*, o navegador Chrome e assim por diante, são amplamente usados e lidam com muitas informações valiosas, armazenando informações confidenciais na memória interna ou no cartão SD do dispositivo.

Analisá-los pode fornecer informações sobre os detalhes do local do usuário, sua comunicação com outras pessoas e muito mais (TAMMA et al., 2018).

Conforme TAMMA et al., 2018, todos os aplicativos armazenam seus dados na pasta `/data/data` por padrão. Os aplicativos também armazenam outros dados no cartão SD, caso seja necessário, solicitando permissão no momento da instalação.

As informações sobre os aplicativos presentes no dispositivo podem ser reunidas inspecionando o conteúdo da pasta `/data/data`. Como alternativa, você pode inspecionar o arquivo "*packages.list*" presente em `/data/system`. Este arquivo contém informações sobre todos os aplicativos junto com seus nomes de pacotes e caminho de dados.

5 INSTALAÇÃO DA ORACLE VM VIRTUAL BOX

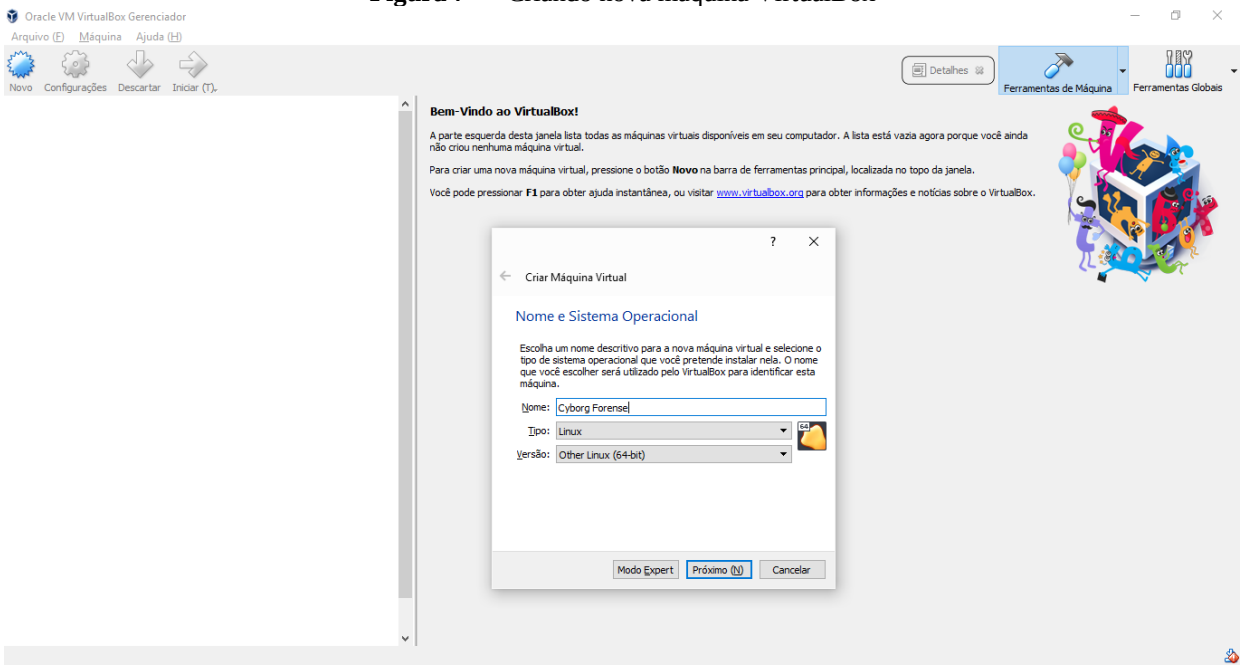
Afim de mostrar as funcionalidades da ferramenta *"AFLogical-OSE_1.5.2"* disponível no *"Cyborg Hawk 1.1"*, está secção visa apresentar a utilização do Android Emulador, com um sistema operacional *Android* 5.1.1.

Para verificação do funcionamento da ferramenta é necessário inicialmente ter uma estação de trabalho forense. Para isso, utilizaremos o sistema operacional *"Cyborg Hawk v1.1"*[23]. Antes da execução do sistema operacional *"Cyborg Hawk v 1.1"*, devemos efetuar o *download* do *software* de máquina virtual mais recente do "*Virtual Box"*[24].

Após o *download*, instale o *software* em sua estação de trabalho forense, em seguida, siga os passos para iniciar a máquina virtual. Localize na "*VM VirtualBox* Gerenciador" a opção "Novo" para cria uma nova máquina virtual, em seguida de um nome de sua preferência e selecione a opção *Linux/Other Linux* (64-bit) conforme a figura 6 abaixo.

[23] Sistema Operacional Disponível para download em: http://cyborg.ztrela.com/cyborg-hawk.
[24] Software para instalação da Máquina Virtual disponível em: https://www.virtualbox.org/wiki/Downloads.

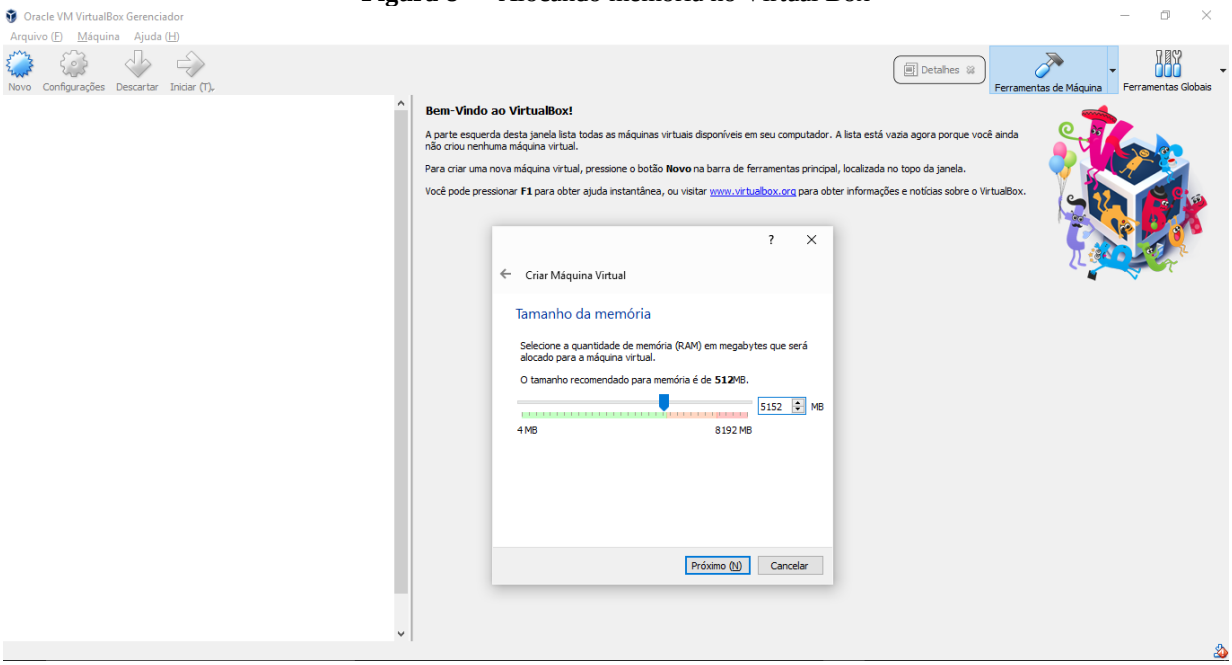
Figura 7 — Criando nova máquina VirtualBox



Fonte: Elaborado pelo próprio autor.

Na próxima etapa aparecerá uma tela informando para selecionar a quantidade de memória (RAM) em megabytes para ser alocada à máquina virtual, o tamanho de 512 MB é o recomendado, no entanto alocaremos a quantidade de memória para 5152 MB, com o intuito de deixar a máquina virtual rápida. Para utilizaremos o Android Virtual Device (AVD), recomenda-se selecionar no mínimo 4GB de memória, em seguida deve-se clicar em “Próximo (N)”.

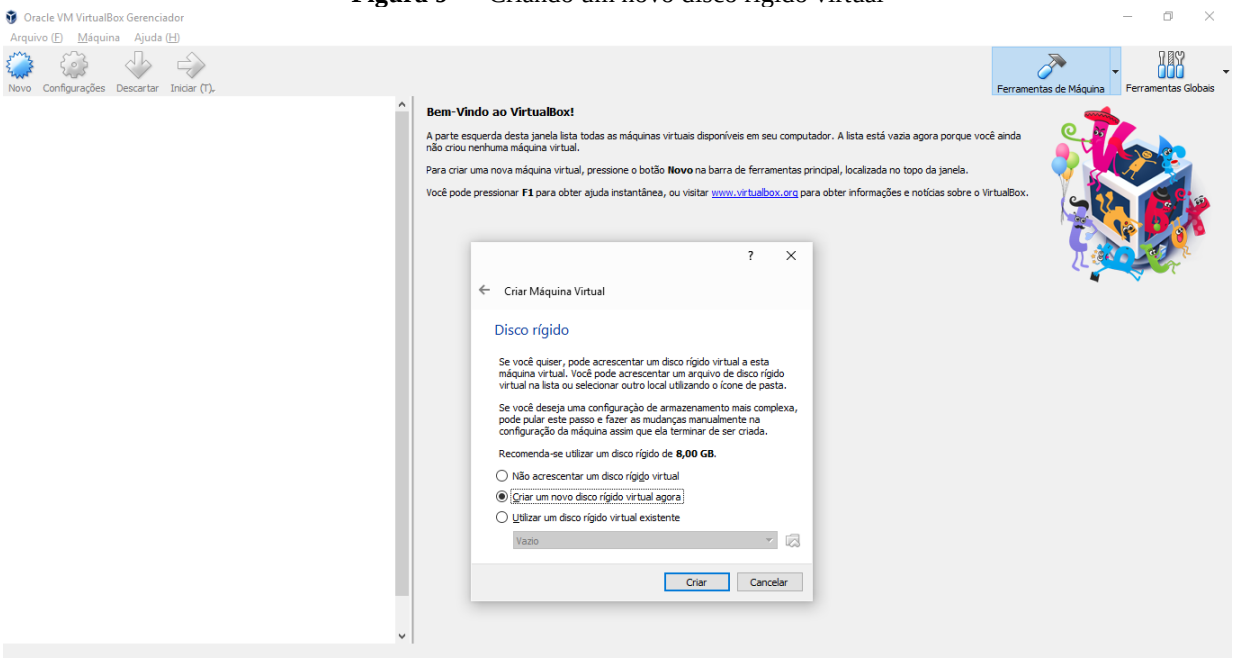
Figura 8 — Alocando memória no Virtual Box



Fonte: Elaborado pelo próprio autor.

Na próxima tela, devemos escolher a opção “criar um novo disco virtual agora”, esta opção possibilita a criação de um disco rígido. Em seguida devemos apertar a opção “Criar”.

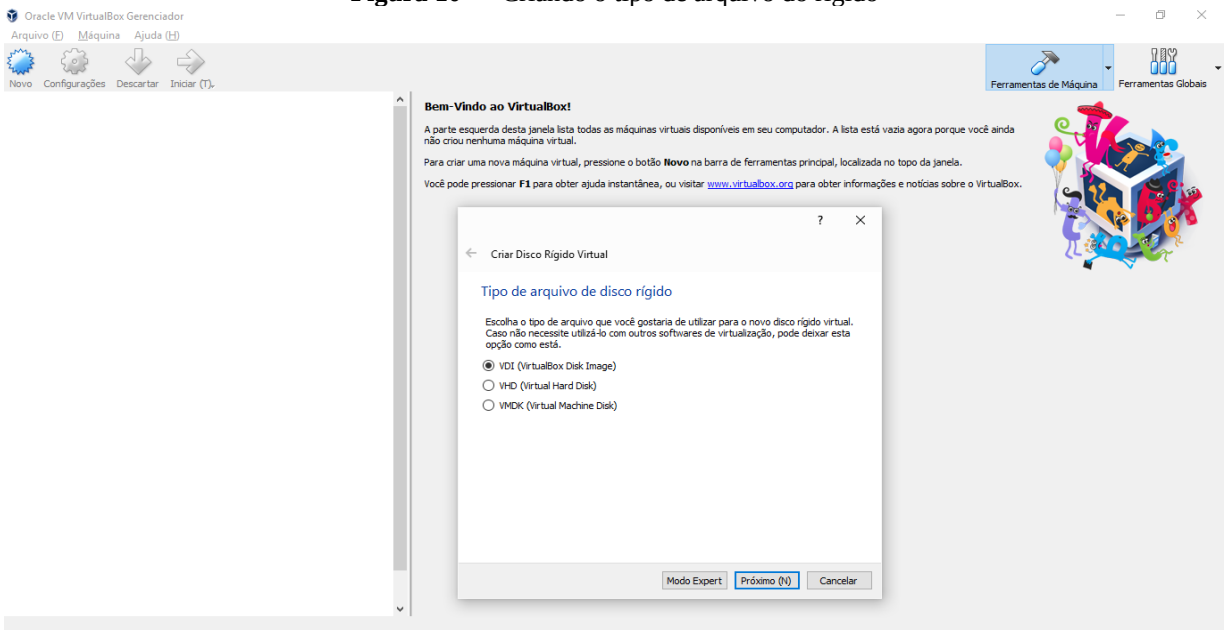
Figura 9 — Criando um novo disco rígido virtual



Fonte: Elaborado pelo próprio autor.

Na tela seguinte, surgirá a opção “Tipo de Arquivo de Disco Rígido”, que solicita a escolha do tipo de arquivo do disco rígido, precisamos escolher a opção VDI (*VirtualBox Disk Imager*) e em seguida clicar na em “Próximo (N)”.

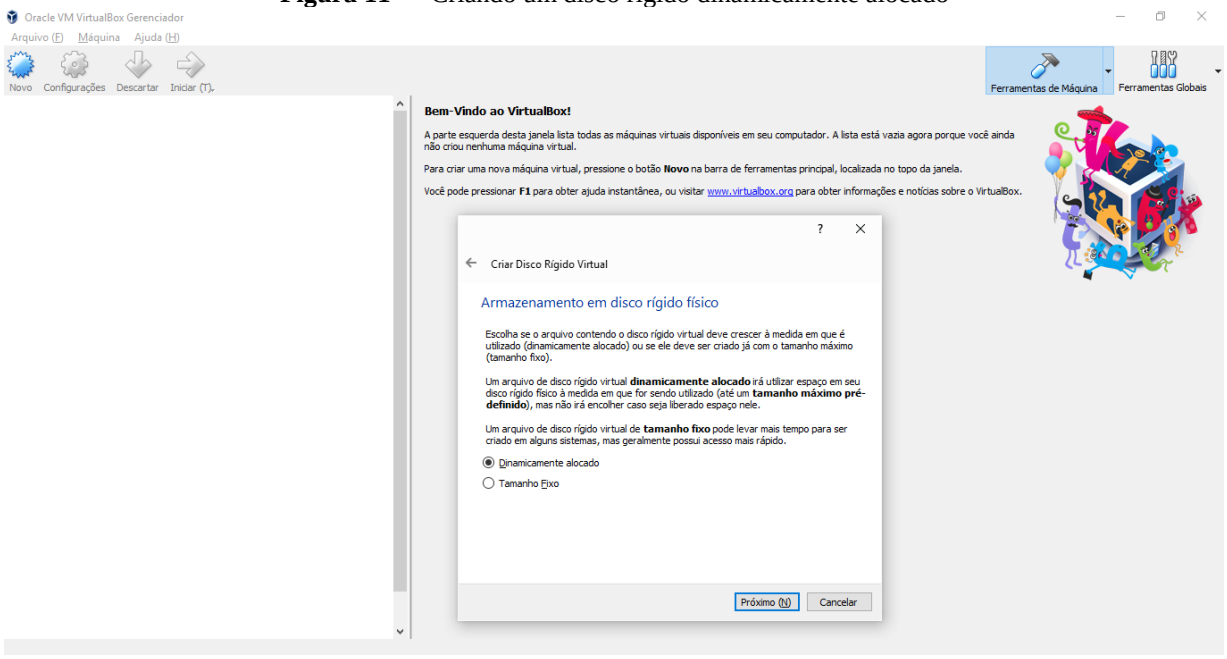
Figura 10 — Criando o tipo de arquivo do rígido



Fonte: Elaborado pelo próprio autor.

Nesta próxima etapa, “Armazenamento em disco rígido físico”, será solicitado que informemos o tipo de armazenamento no disco rígido, escolha a opção “dinamicamente alocado”, que criará um disco rígido virtual, que utiliza espaço em seu disco físico à medida que for sendo utilizado, em seguida deve-se clicar em “Próximo (N)”.

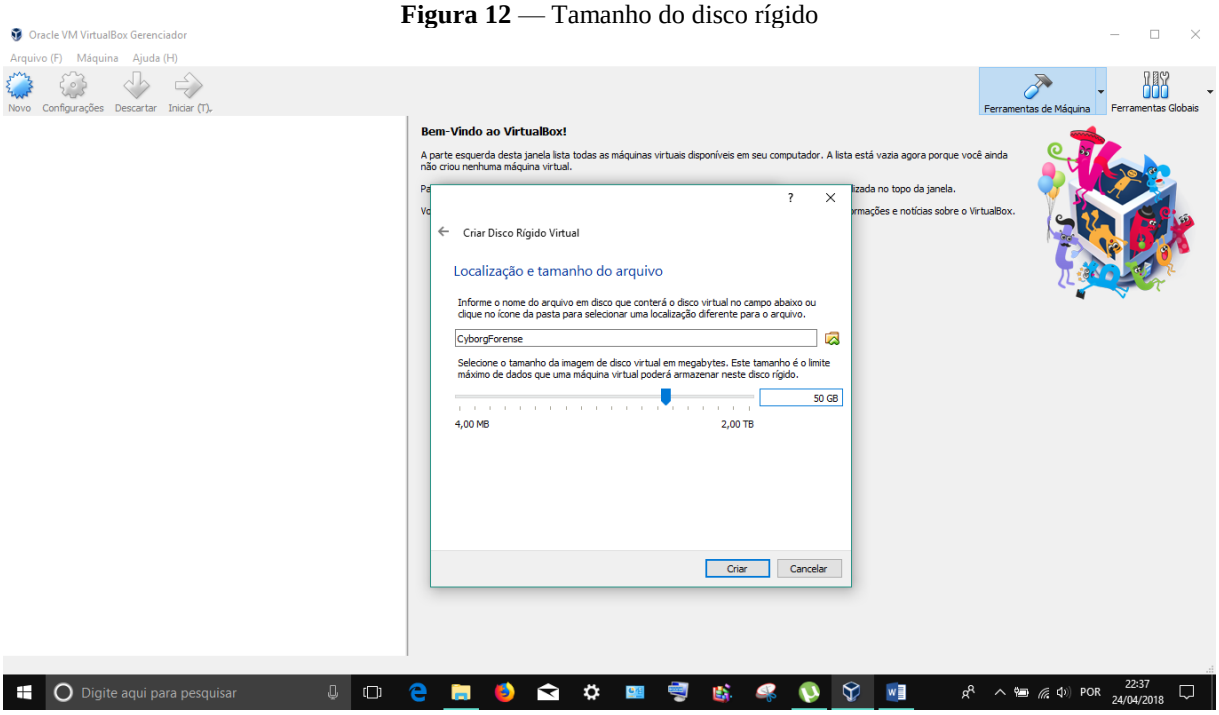
Figura 11 — Criando um disco rígido dinamicamente alocado



Fonte: Elaborado pelo próprio autor.

Neste momento, deve-se ajuste o tamanho do disco rígido em megabyte para alocação, onde o recomendado é de 50 gigas. Ao termino, clique em “Criar”.

Figura 12 — Tamanho do disco rígido

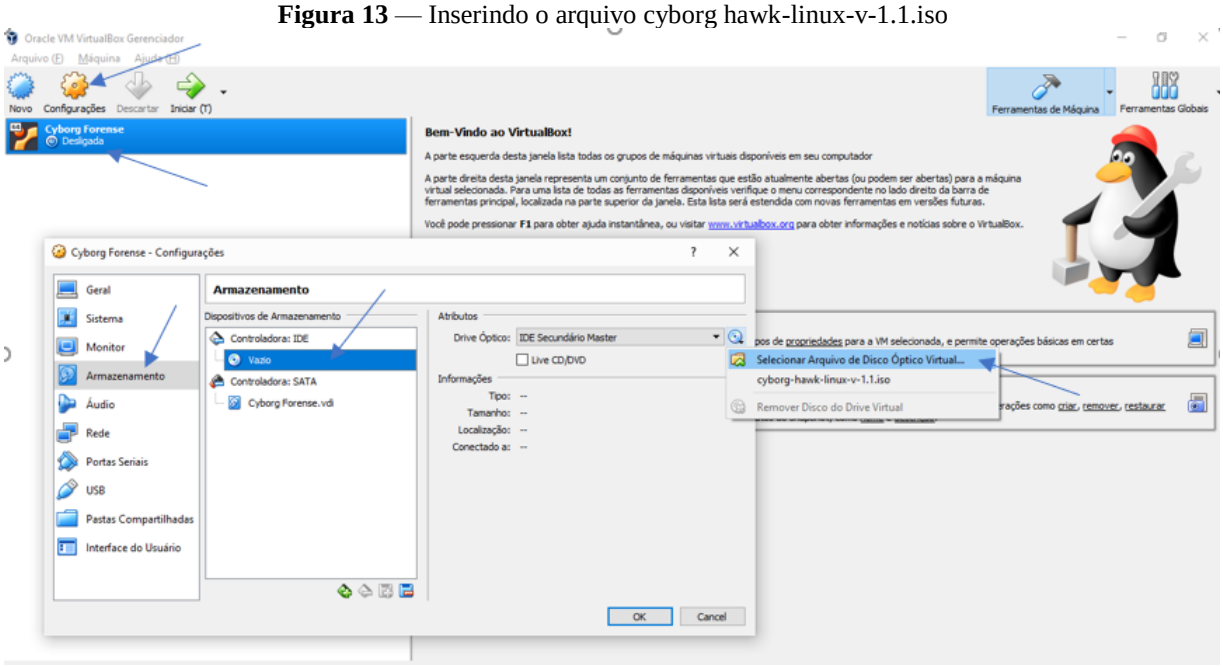


Fonte: Elaborado pelo próprio autor.

Após realizar as configurações necessárias, devemos inserir o *Cyborg Hawk* à máquina virtual recém-criada, isto é semelhante a colocação de um CD (*Compact Disc*) ou DVD (*Digital Versatile Disc*) para inicializar e instalar um novo sistema operacional.

Para isso devemos selecionar a máquina virtual criada “Cyborg Forense”, ir em configurações no topo da tela *VM VirtualBox* Gerenciador e depois clicar no menu “Armazenamento”, em seguida clicar na opção “Vazio” do “Controlador IDE”. Feito isso, vá até “Atributos” clique no ícone de CD para selecionar de seu computador o Arquivo de Disco Óptico Virtual e depois de um clique em “OK”, com segue a figura 12 abaixo.

Figura 13 — Inserindo o arquivo cyborg hawk-linux-v-1.1.iso

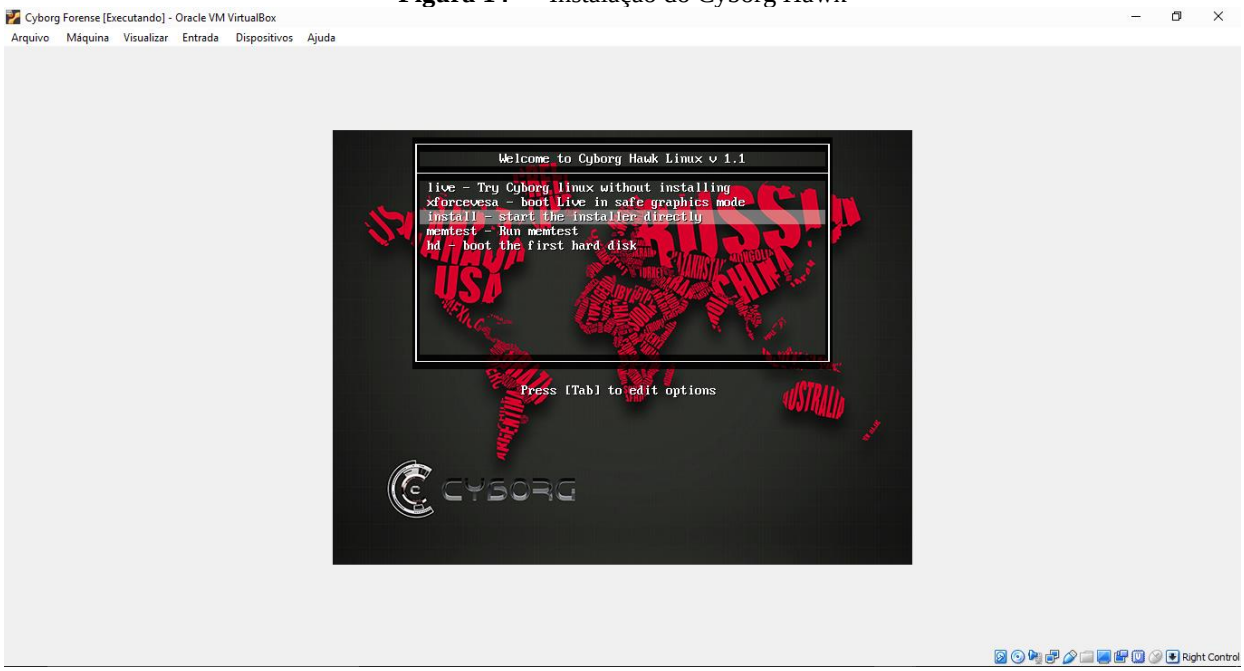


Fonte: Elaborado pelo próprio autor.

6 INSTALANDO O CYBORG-HAWK 1.1

Neste instante, com a máquina do “Cyborg Forense” criada e devidamente configurada, podemos seleciona-la e clicar em “Iniciar”, na tela principal do VM *VirtualBox* Gerenciado, para começar o processo de instalação. Após iniciada, irá aparecer algumas opções de instalação, devemos escolher a opção “install – start the installer directly” e pressionar a tecla “Enter”, como mostra a figura 13 abaixo.

Figura 14 — Instalação do Cyborg Hawk



Fonte: Elaborado pelo próprio autor.

Logo após, será necessário durante o processo de instalação, efetuar algumas configurações, tais como: inserir informação de Linguagem, Fuso-horário e Configurações de relógio e *Layout* do teclado.

6.1 Configurando e atualizando o SDK Manager

O sistema operacional "Cyborg Hawk", vem previamente com o *Android SDK Manager* instalado, este foi o motivo primordial pela escolha do sistema operacional, além de sofrer atualizações constantes. Contudo, é necessário atualizar a plataforma do *Android SDK Manager*.

Para prosseguirmos com a atualização, vá até o Menu *Applications*, em seguida acesse os sub menus: *Cyborg* > *Mobile Security* > *Development Tools* > *Android SDK Manager*.

O Android SDK Manager oferece ferramentas, e outros componentes do SDK, necessários para desenvolver aplicativos, no entanto, por recomendação, devemos considerar cuidadosamente as seguintes ferramentas na guia SDK Tools:

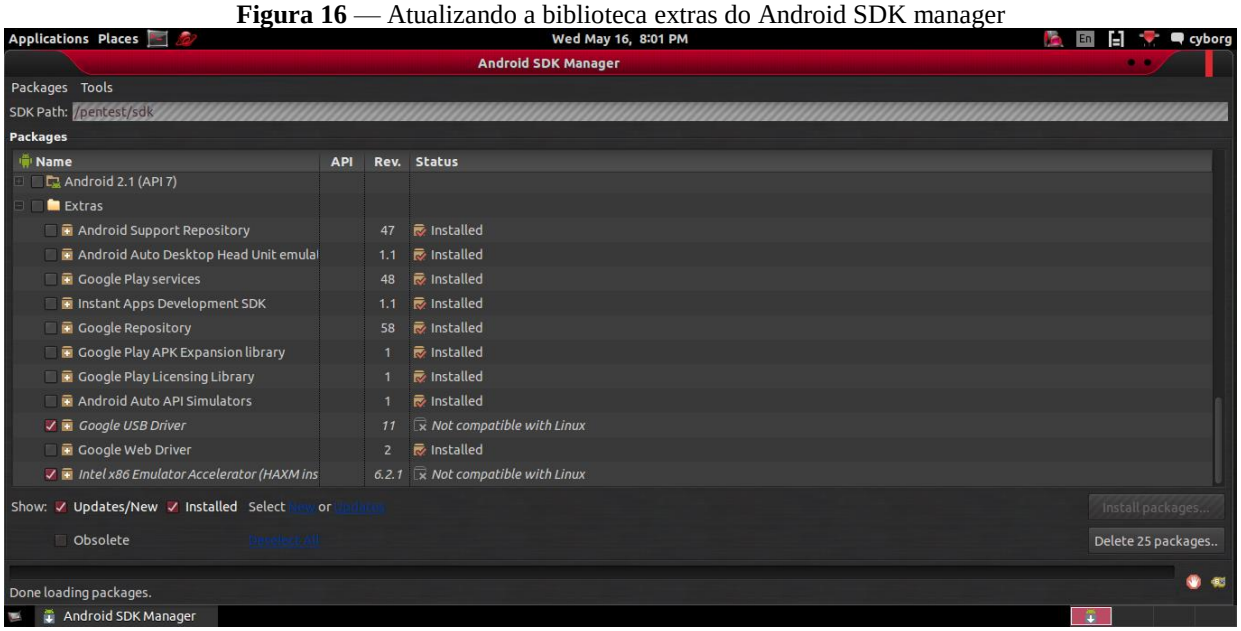
a) Android SDK Build-tools: inclui ferramentas para compilar aplicativos Android.
b) Android SDK Platform-tools: inclui diversas ferramentas necessária para plataforma Android incluindo a ferramenta adb.
c) Android SDK Tools: inclui ferramentas essenciais como o Android Emulador. (ANDROID, 2018).

Para atualizar o *SDK Manager*, deve-se marcar na opção *Packages* do *SDK Manager*, as caixas ao lado do pacote *Tools*. Por opção, iremos marcar todas as caixas do pacote *Tools* do Pacote Extras, em seguida, clicar em *Accept License* > *install* e aguarde a atualização, como segue a figura 15 abaixo:

Figura 15 — Atualizando as ferramentas do Android SDK manager

Fonte: Elaborado pelo próprio autor.

Figura 16 — Atualizando a biblioteca extras do Android SDK manager

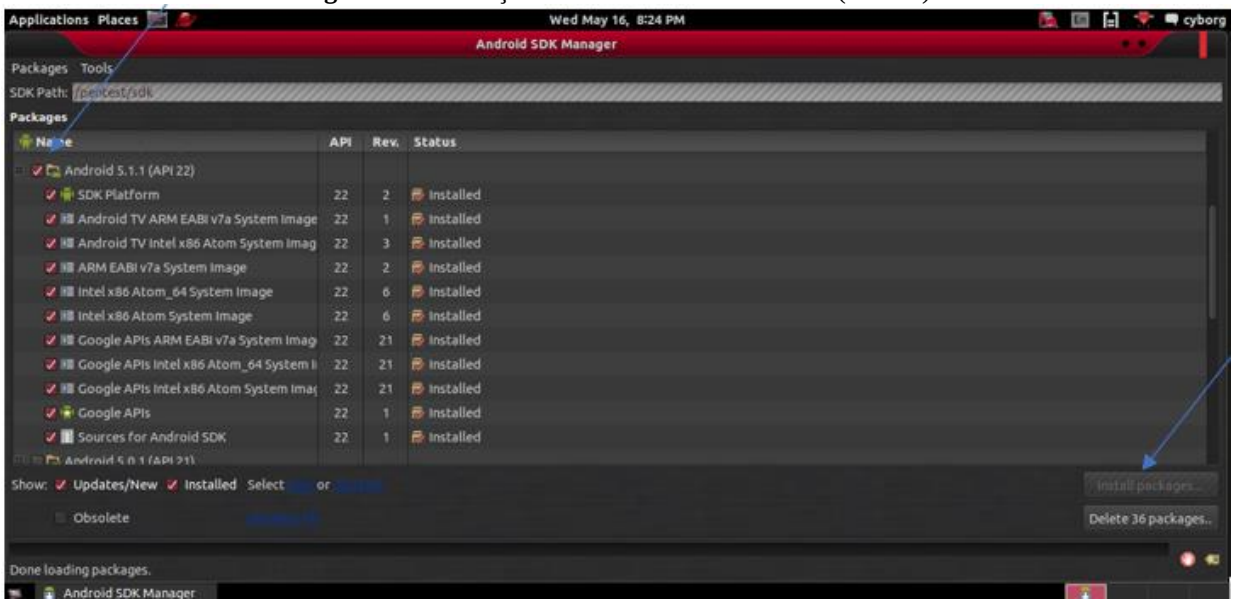


Fonte: Elaborado pelo próprio autor.

6.2 Emulando um dispositivo virtual

Para criar e gerenciar um dispositivo virtual deve-se executar o *Android SDK Manager*, será exibido uma lista de *Packages* com diferentes versões de *Android*. Para os testes que serão realizados, vamos escolher a versão do *Android* 5.1.1 (API 22). Desta maneira, iremos marca a caixa ao lado da versão do hardware escolhido e em seguida clicar em *Install Packages*, como mostra a imagem 17 abaixo:

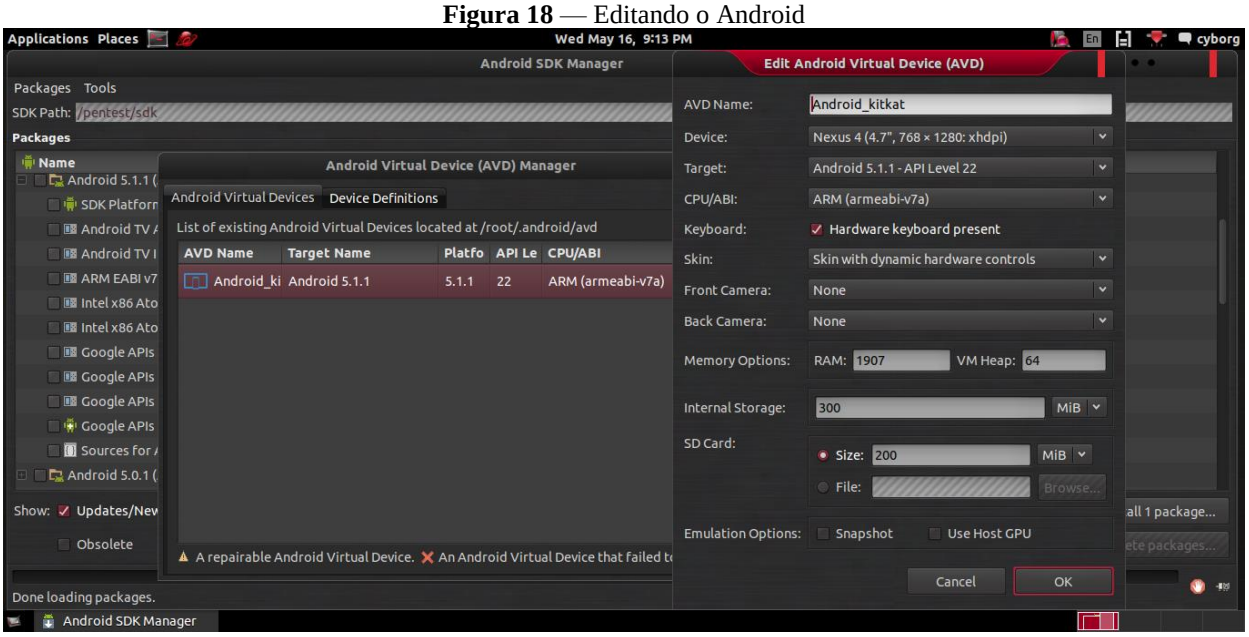
Figura 17 — Seleção da versão do Android 5.1.1 (API 22)



Fonte: Elaborado pelo próprio autor.

Feito isto, devesse clicar em *Tools* novamente, abrirá um tela com uma lista de dispositivos virtuais criados, deve-se selecionar o Android que foi alocado e clicar em *Edit*, para criar um perfil de *hardware* adequado, neste momento é possível definir características como: nome do AVD, o tipo de dispositivo que será usado, o *Target* que é a versão do hardware que foi baixado, CPU e outras opções avançadas, como utilização do *host* da GPU de sua máquina física.

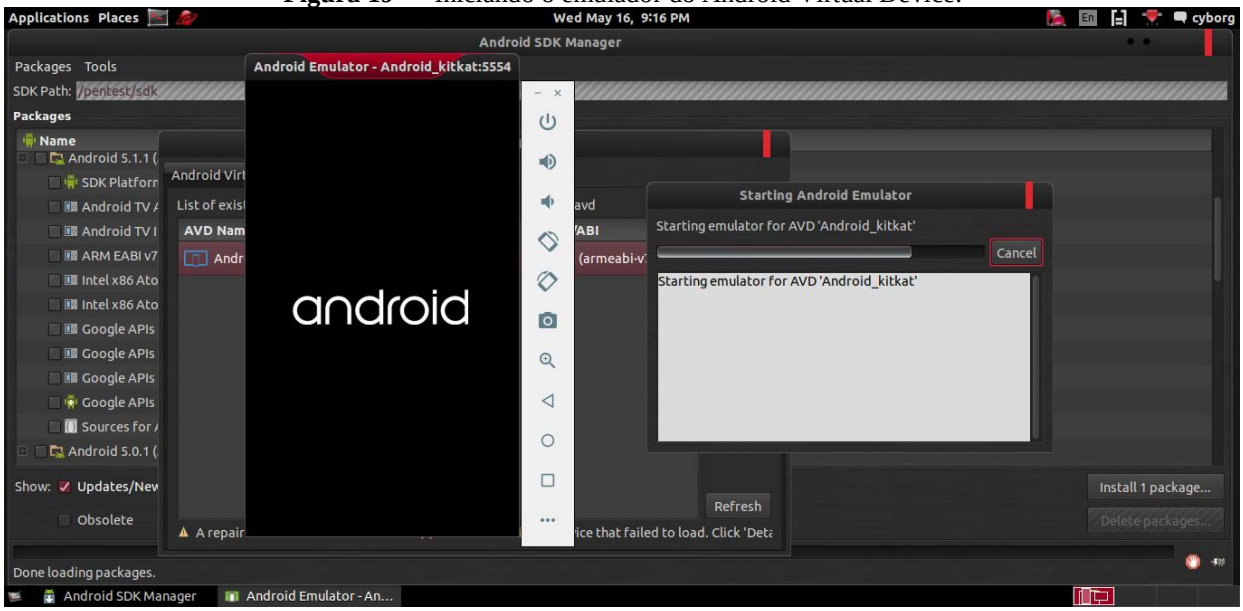
Figura 18 — Editando o Android



Fonte: Elaborado pelo próprio autor.

Caso se deseje alterar as configurações de seu AVD, vá até o menu principal do gerenciador de AVDs, e clique na opção “Editar” para realizar as mudanças necessárias. Uma vez criado seu AVD, precisa-se clicar em “Ok” escolher seu AVD criado e depois clicar em “Start”, em seguida escolher as opções aplicáveis em “*Launch Options*” e clicar em “*Launch*”. Feito isto, o Android emulador será iniciado, assim como mostra a figura 19 abaixo:

Figura 19 — Iniciando o emulador do Android Virtual Device.



Fonte: Elaborado pelo próprio autor.

Portanto, caso não ocorra nenhum problema na inicialização do *AVD Android Lollipop* criado, a ferramenta abrirá na tela, como mostra a figura 20 abaixo:

Figura 20 — Inicialização do emulador

Fonte: Elaborado pelo próprio autor.

6.3 Verificando e executando o android debug bridge (ADB)

O Android Debug Bridge (ADB) [25] é uma ferramenta de linha de comando que permite a comunicação com uma instancia de emulador ou com um dispositivo Android conectado, ele possibilita uma variedade de ações, como a possibilidade de instalar e depura aplicativos, e fornece um *Shell Unix* que pode ser usado para executar diversos comandos em um emulador ou dispositivo conectado. A ferramenta ADB está localizada no seguinte diretório (ANDROID, 2018):

```
android_sdk/platform-tools/
```

Quando você inicia um cliente do ADB, ele primeiro verifica se há um processo de servidor do ADB em execução, caso não haja um processo em execução ele inicia este processo (ANDROID, 2018).

6.3.1 Verificando e Executando o *ADB Devices*

Assim que o servidor é inicializado ele vincula à porta TCP 5037 local e escuta comandos enviados do cliente adb. Todos os clientes que utilizam a porta 5037 para comunicação com o servidor adb. Por sua vez, o servidor configura conexões com todas as instâncias de emulador/dispositivo em execução. O servidor localiza as instâncias de emulador varrendo as portas impares no intervalo de 5555 a 5585, utilizado por emulador dispositivo. (ANDROID, 2018)

Onde o servidor encontrar um daemon do adb, ele configurará uma conexão com a porta. É importante notar que cada instância de emulador/dispositivo adquire um par de portas sequenciais, uma porta para conexões de console e a outra para conexões do adb. (ANDROID, 2018).

Desta maneira, é possível verificar se o emulador foi reconhecido, para isso; vá até o terminal e execute o comando:

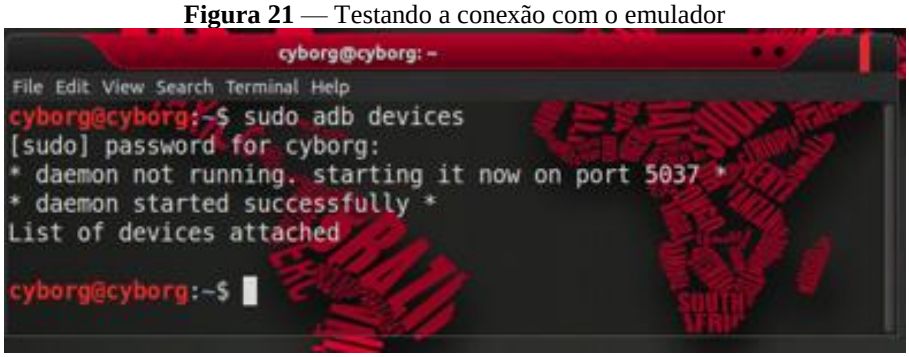
```
sudo adb devices.
```

Na tela, deve aparecer a seguinte mensagem: *"List devices attached emulator-5556 devices"*.

Caso não apareça esta mensagem ou apareça "*off line*", deve-se esperar 60 segundos, enquanto o reconhecimento ainda não foi feito pela máquina criada "*CyborgForense*".

[25] Para mais informações acesse o site: https://developer.android.com/studio/command-line/adb

Figura 21 — Testando a conexão com o emulador



Fonte: Elaborado pelo próprio autor.

Após ter passado o tempo de necessário para o reconhecimento, repita a execução do comando "`sudo adb devices`" novamente.

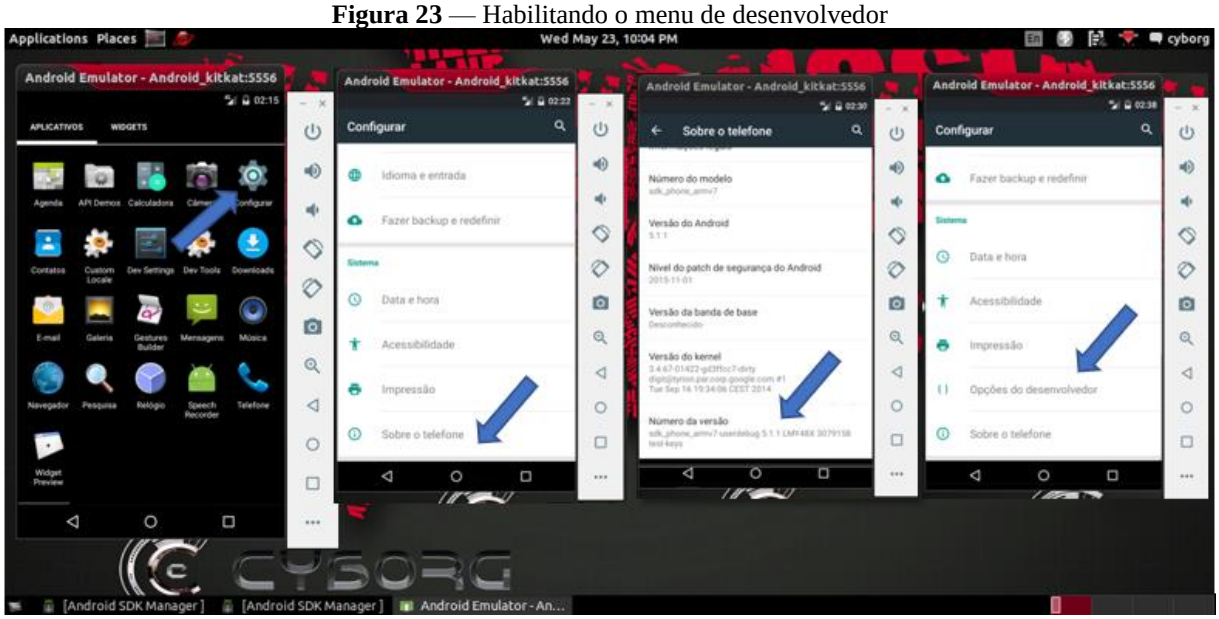
Figura 22 — Inicialização do emulador

Fonte: Elaborado pelo próprio autor.

6.3.2 Ativando a depuração do ADB do dispositivo emulado

Para usar o ADB com um dispositivo emulado ou conectado via USB, você deve ativa a opção de depuração USB. Antes disso, você deve ativar o modo desenvolvedor nas configurações de seu dispositivo emulado, como montra a figura 23 logo abaixo (ANDROID, 2018).

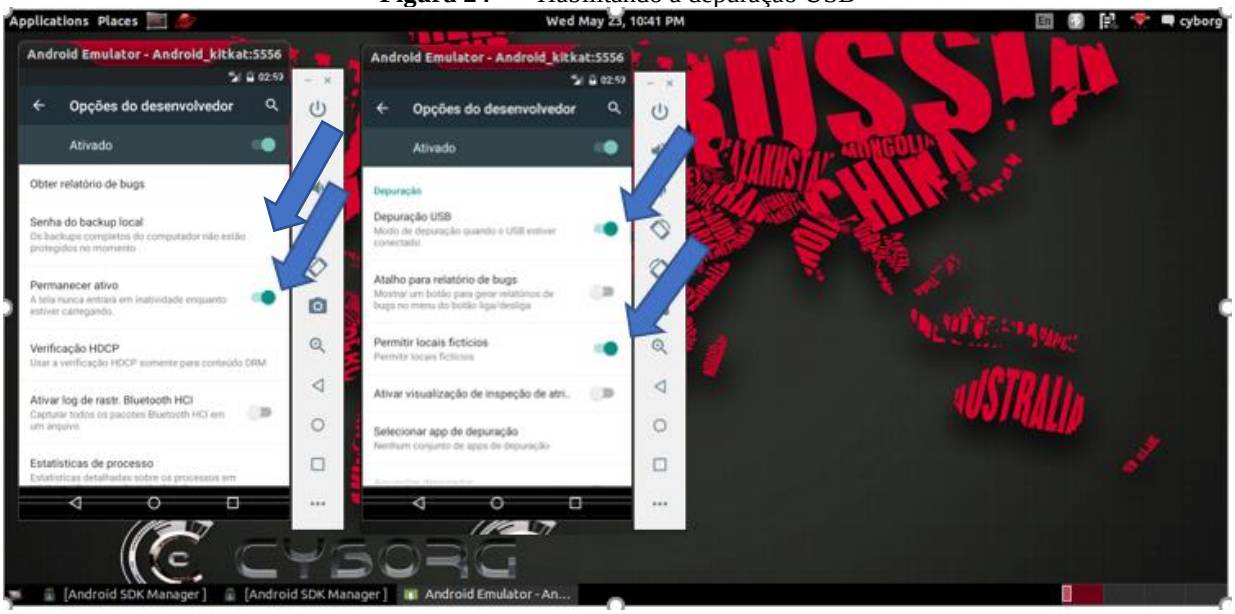
Figura 23 — Habilitando o menu de desenvolvedor



Fonte: Elaborado pelo próprio autor.

Após ter habilitado as Opções de Desenvolvedor, devemos habilitar a Opção *Permanecer ativo > Depuração USB > Permitir Locais Fictícios.* A opção *Permanecer Ativo*, permite que a tela nunca entre em inatividade, enquanto a tela não estiver carregando; já a opção *Depuração USB*, possibilitará conexão ao dispositivo emulado ou conectado via USB e por fim; a opção *Permitir locais Fictício*, simulará locais fictícios no dispositivo emulado. Habilitando-se os módulos do aparelho, ele estará preparado para ceder a extração, como segue a imagem 24 abaixo.

Figura 24 — Habilitando a depuração USB

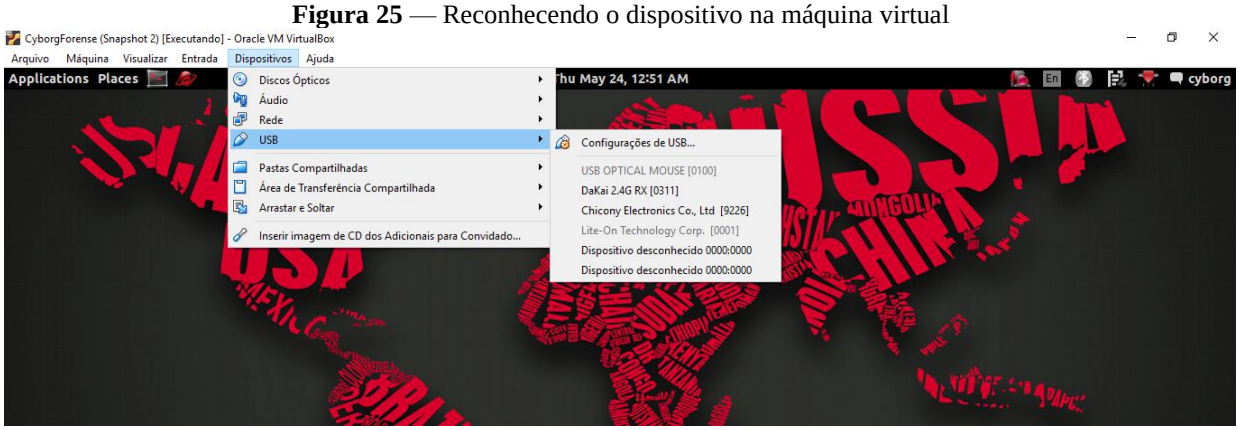


Fonte: Elaborado pelo próprio autor.

7 UTILIZANDO AFLOGICAL OSE PARA EXTRAÇÃO DE DADOS NO DISPOSITIVO EMULADO

Para instalar o *AFLogical-OSE_1.5.2* é necessário que o dispositivo seja reconhecido pela máquina virtual "*CyborgForense"*, então devemos ir na opção: Dispositivo da "*Oracle VM VirtualBox*" e nos certificar que o mesmo foi reconhecido, como mostra a imagem 25 abaixo.

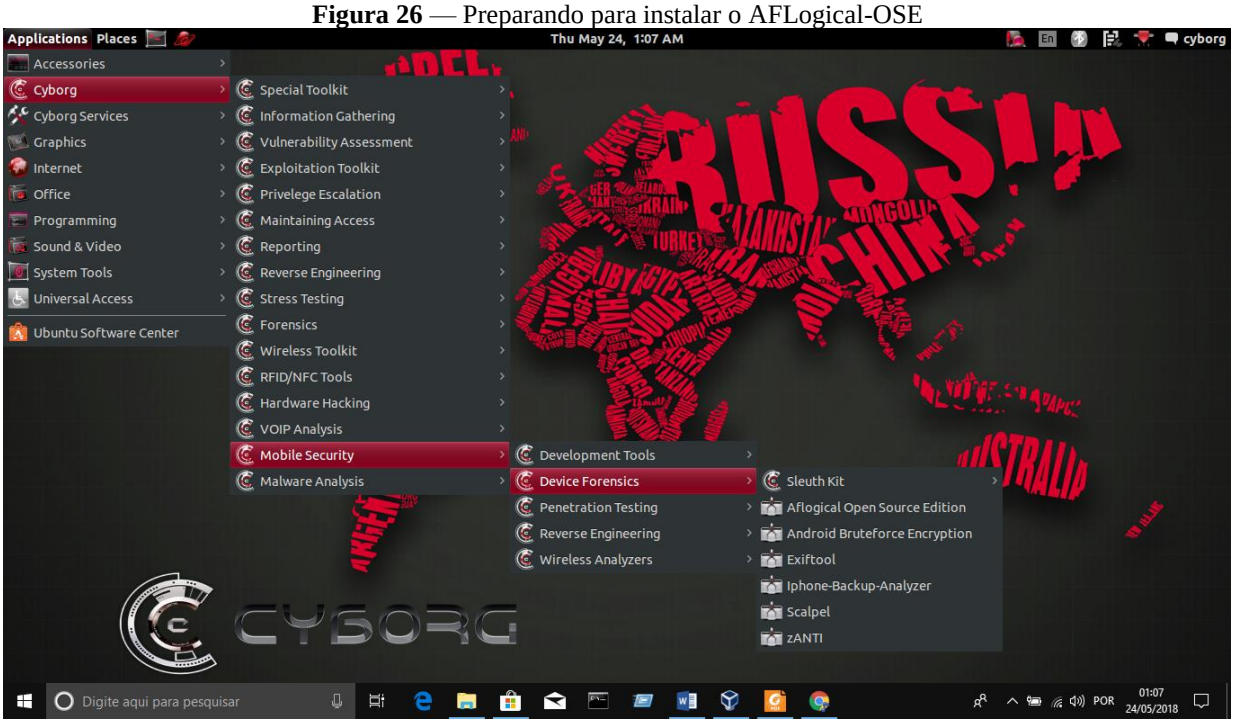
Figura 25 — Reconhecendo o dispositivo na máquina virtual



Fonte: Elaborado pelo próprio autor.

Após verificar que o aparelho emulado foi conectado é necessário isola-lo de qualquer forma de comunicação externa, para isso coloque o mesmo em "Modo avião". Feito isso, deve-se ir no menu de opções, clicar em: *Applications* do *CyborgForense*, em seguida encontra as opções *Cybor > Mobile Security > Device Forensics >Aflogical Open Source Edition*.

Figura 26 — Preparando para instalar o AFLogical-OSE



Fonte: Elaborado pelo próprio autor.

Agora nesta etapa abrirá uma tela de terminal solicitando a senha padrão de usuário root, desta maneira informe a senha. Verifique se a ferramenta forense pode ser comunica com seu dispositivo Android, desta forma; execute o seguinte comando no terminal: `ls -l`.

Aparecerá na tela uma lista de dispositivo conectados da seguinte forma.

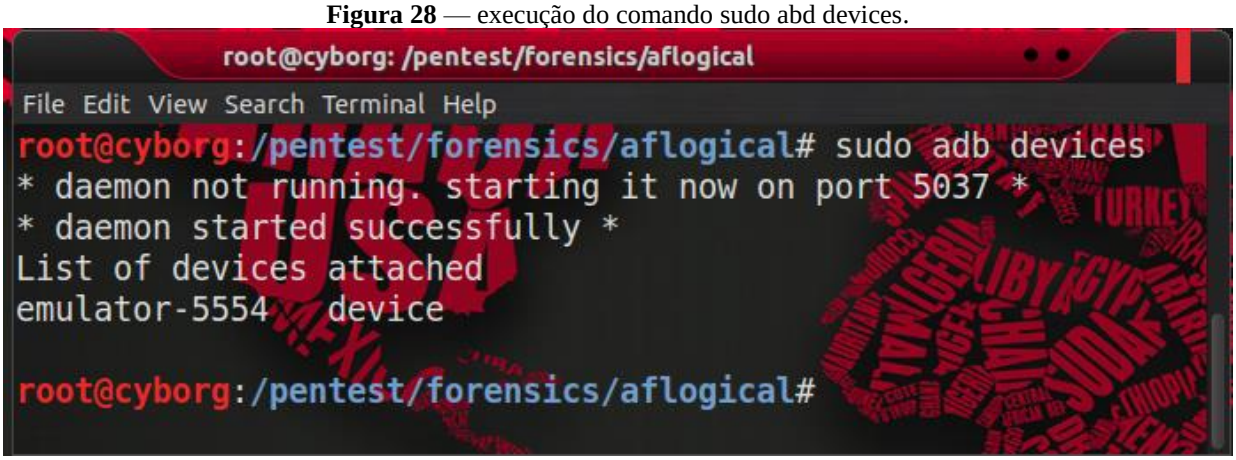
Figura 27 — conectando o dispositivo via ADB device.

Fonte: Elaborado pelo próprio autor.

Logo após, devemos inserir o AFLogical-OSE_1.5.2 para dentro do dispositivo emulado através da ferramenta ADB, utilizando o comando: "`sudo adb devices`".

Aparecerá a mensagem "*List of Devices attached emulator-5554 Device*". como mostra a figura 28, logo abaixo:

Figura 28 — execução do comando sudo abd devices.



Fonte: Elaborado pelo próprio autor.

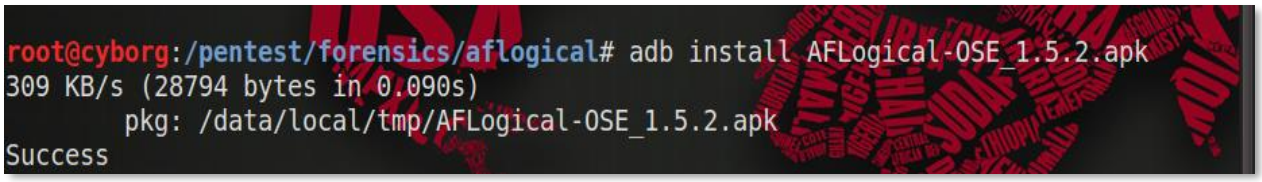
A mensagem que é resultado da execução do comando "`sudo adb devices`", informa que o processo "daemon" não está rodando. O é daemon é executado como um processo de segundo plano em cada instância de emulador ou dispositivo (ANDROID, 2018).

Neste momento, precisamos fazer a instalação do AFlogical OSE para isso, digite ainda no terminal o seguinte comando:

```
adb install AFLogical-OSE_1.5.2.apk
```

Apos o comando de instalção aparecerá o seguinte mensagem, como mostra a figura abaixo:

Figura 29 — Instalação do AFlogical Sucess.



Fonte: Elaborado pelo próprio autor.

Agora devemos ir até o dispositivo emulado e abrir o *App AFLogical-OSE_1.5.2* instalado, executa-lo e escolher os dados para extração, como segue a imagem 30, logo abaixo:

Figura 30 — Extração dos dados do aparelho celular.

Fonte: Elaborado pelo próprio autor.

7.1 Extração dos dados do cartão SD

Nesta etapa, devemos extrair os dados do cartão SD para a máquina “CyborgForense”. Primeiro devemos criar um diretório, como o seguinte comando no terminal:

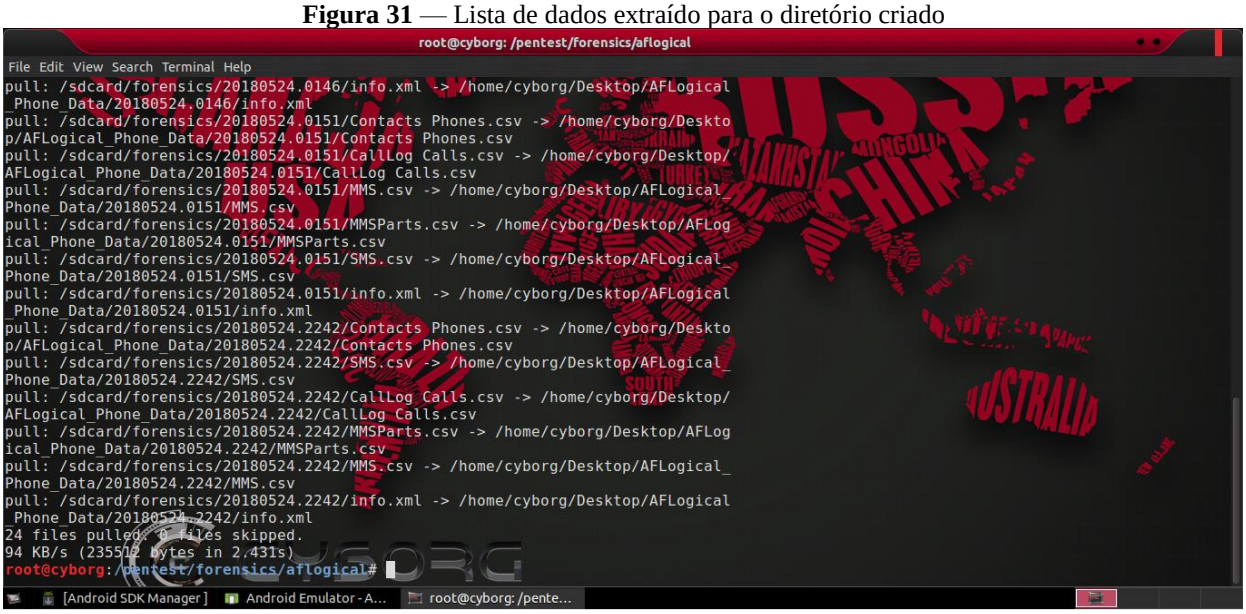
```
mkdir ~/Desktop/AFLogical_Phone_Data”
```

E efetuar a extração dos dados com o comando abaixo, para o diretório criado:

```
adb pull /sdcard/forensics/ ~/Desktop/AFLogical_Phone_Data
```

Em seguida aparecerá a quantidade de arquivos retirados bem como os arquivo que foram ignorados, como segue a figura 31 abaixo:

Figura 31 — Lista de dados extraído para o diretório criado



Fonte: Elaborado pelo próprio autor.

Note: você dever ter um cartão instalado no seu disposto para extrair os dados para o diretório. No entanto, no momento da edição do AVD, definimos uma memória de 200 MB para o dispositivo emulado. Porém, como o dispositivo está sendo emulado para fins de teste da Aplicação *AFLogical-OSE*, ele não possui dados algum para efeito de visualização do que foi extraído, mas, a extração dos dados para o diretório ocorreu de forma eficaz, como mostra o resultado acima na imagem 32 acima:

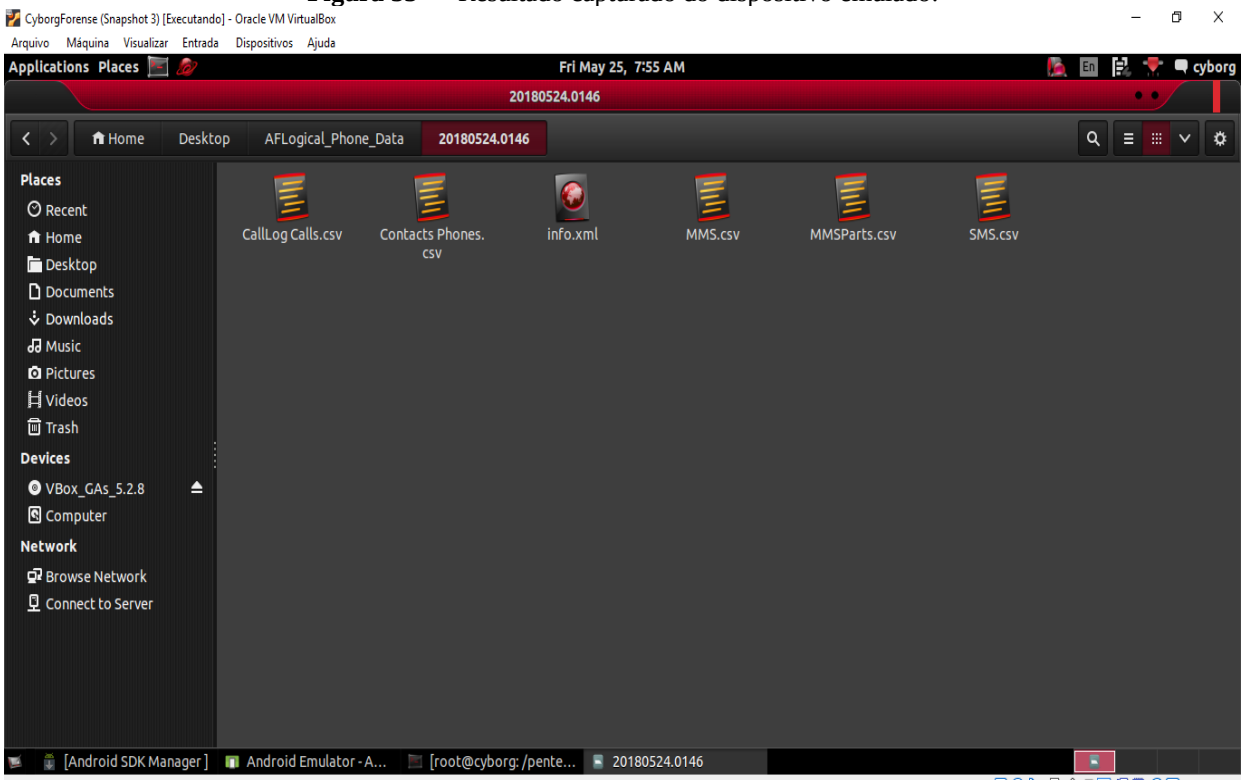
Figura 32 — Extração dos dados puxados e ignorados

Fonte: Elaborado pelo próprio autor.

Por fim, pode-se consultar os dados que fora extraído para o diretório: `cyborg@cyborg: ~/Desktop/AFLogical_Phone_Data$`

Este diretório contém uma pasta nomeado com data e horário da extração, contendo dados como registro de chamas, contatos, MMS/SMS e informações do dispositivo em formato SCV, como mostra imagem 33 abaixo:

Figura 33 — Resultado capturado do dispositivo emulado.

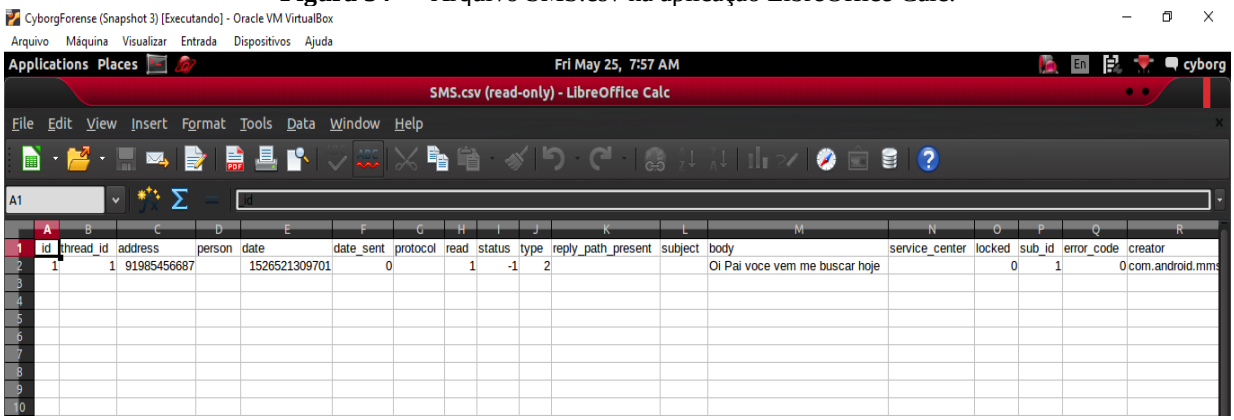


Fonte: Elaborado pelo próprio autor.

Abaixo podemos ver os resultados de registros de chamadas em uma planilha do LibreOffice, como mostra a imagem 34.

Este arquivo mostra o corpo de uma comunicação via SMS de uma mensagem entre o emissor e o receptor.

Figura 34 — Arquivo SMS.csv na aplicação LibreOffice Calc.



Fonte: Elaborado pelo próprio autor.

8 UTILIZANDO APARELHOS CONECTADOS VIA USB

Para os testes realizados aqui, será considerada como objetivo a extração das informações mais relevante, julgado pelo analista pericial, sejam elas registros de chamadas, e-mail, imagem, vídeos e tudo que possa ser utilizado para suprir o processo investigativo. Será utilizado apenas o ambiente criado na "*VM VirtualBox*", junto com o sistema operacional "*CyborgForense*", utilizado três smartphones com diferentes versões do sistema operacional *Android*.

As características dos equipamentos encontram-se no quadro abaixo.

Quadro 4 — Descrição dos testes propostos

	Característica dos Smartphones				
RESULTADOS	Ligado	Bloqueio	Acesso a depuração	Super usuário	Descrição
1	Sim	Não	Não	Não	ASUS Zenfone 4Max, Modelo Asus_X00ID, Android versio 7.1.1.
2	Sim	Sim	Sim	Não	Motorola, Modelo Moto C Plus, Android 7.0
3	Sim	Sim	Sim	Não	Samsung Galaxy J1 (2016), Modelo SM – J120H

Fonte: Elaborada pelo autor

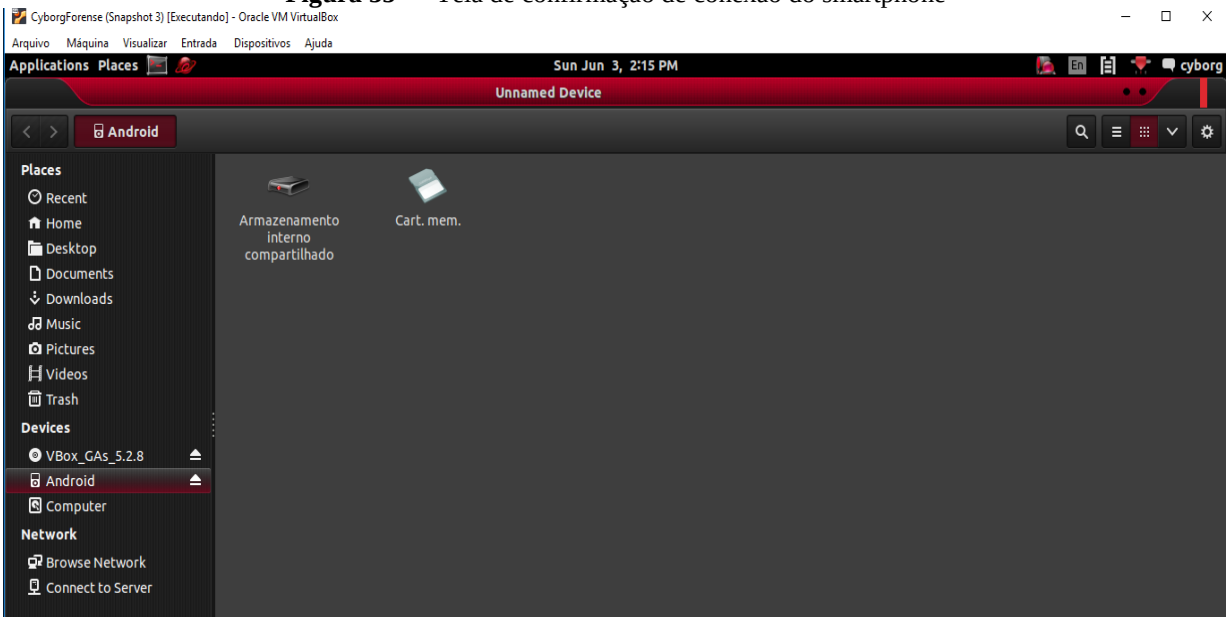
8.1 Resultado 1 – Aparelho ligado, desbloqueado e sem permissões de super usuário

Este cenário tem como objeto apresentar uma situação de exame de um smartphone, de um usuário normal, que utiliza apenas as aplicações básicas, sem controle de acesso ativo e tampouco permissões de super usuário instaladas. Considera-se que no momento da apreensão do dispositivo o mesmo não se encontra desligado ao ter sido isolado da rede, e sendo colocado em modo avião. O smartphone utilizado neste senário é o ASUS Zenfone 4Max, Modelo Asus_X00ID, ele permite a remoção do cartão SD para extração dos dados, o que facilitaria sua análise e gerenciamento do cartão de memória e sua eventual clonagem.

8.1.1 Instalação do Oracle VM VirtualBox Extension Pack

Para a análise do cartão de memória, foi instalado na máquina *Windows* hospedeira a extensão, *Oracle VM VirtualBox Extension Pack*[26]. Essa extensão de pacote possibilita o reconhecimento de dispositivos USB 2.0 e USB 3.0, inseridos na máquina física com a máquina virtual. Feito a instalação deste pacote, devemos conectar o dispositivo via UBS, logo em seguida vá até a máquina virtual "*CyborgForense"*, clique em Dispositivo > USB, observe se está aparecendo o dispositivo conectado e de um clique no mesmo, feito isto abrirá a tela como mostra a figura 35 abaixo:

Figura 35 — Tela de confirmação de conexão do smartphone



Fonte: Elaborado pelo próprio autor.

.

Após o smartphone ter sido devidamente conectado na máquina virtual, o mesmo foi desligado e logo em seguida foi retirado o cartão de memória do dispositivo e inserido em um adaptador microSD, na tentativa de realizar uma cópia *bit* a *bit* para análise.

A máquina virtual consegui reconhecer o dispositivo Android conectado via UBS, mas não foi possível o reconhecimento de adaptadores de cartão de memória na máquina "*CyborgForense"*, desta forma a solução encontra foi a realização de uma inspeção manual por meio de navegação nas pastas do dispositivo. Mas, ante da inspeção manual dos arquivos contido no cartão de memória, será utilizado neste primeiro teste apenas a ferramenta *AFLogical_OSE*.

[26] Download disponível em: https://www.virtualbox.org/wiki/Downloads

Após certificarmos que o aparelho foi reconectado via USB e que foi isolado de qualquer forma de comunicação posto em modo avião, que as opções de “Depuração Via USB” “Permanecer Ativo” estão habilitadas, o aparelho se encontra devidamente configurado para a análise, como mostra a figura 35 abaixo:

Figura 36 — Habilitando o dispositivo móvel

Fonte: Elaborado pelo próprio autor.

Após realizarmos as devidas configurações devemos ir até a ferramenta *AFlogical* no sistema “*CyborgForense”* e seguir os seguintes passos: cyborg > Mobile Security > Devices Forensics > AFlogical Open Source Edition. A ferramenta abrirá em um terminal, solicitando para inserimos a senha de usuário root. Logo após inserirmos a senha, digite os seguintes comandos:

```
root@cyborg:/pentest/forensics/aflogical# ls
```

Aparecerá a mensagem informando:

“AFLogical-OSE_1.5.2.apk GPL README.txt”.

Agora devemos inserir o AFlogical_OSE para dentro do dispositivo *ASUS Zenfone 4Max, Modelo Asus_X00ID*, informando o comando:

```
root@cyborg:/pentest/forensics/aflogical# sudo adb devices
```

Aparecerá esta mensagem com a lista de dispositivos anexados.

List of devices attached

H9AXB602D407DD5   device

Neste momento faça a instalação do AFlogical.apk no dispositivo, digitando o comando:

```
root@cyborg:/pentest/forensics/aflogical# adb install AFLogical-OSE_1.5.2.apk
```

Surgirá a seguinte mensagem

309 KB/s (28794 bytes in 0.090s)

Success

Como mostra a mensagem acima, o aplicativo AFlogical _OSE foi instalado com êxito.

Agora devemos ir até o aplicativo e abri-lo. Será solicitado que você escolha os dados que será extraído do dispositivo, como mostra a figura 37 abaixo.

Figura 37 — Seleção e extração de dados do dispositivo

Fonte: Elaborado pelo próprio autor.

Nesta etapa, devemos extrair os dados para a máquina "*CyborgForense"*. Entretanto, deve-se criar um diretório na máquina "*CyborgForense"* para realizarmos o deposito dos dados conforme o comando abaixo:

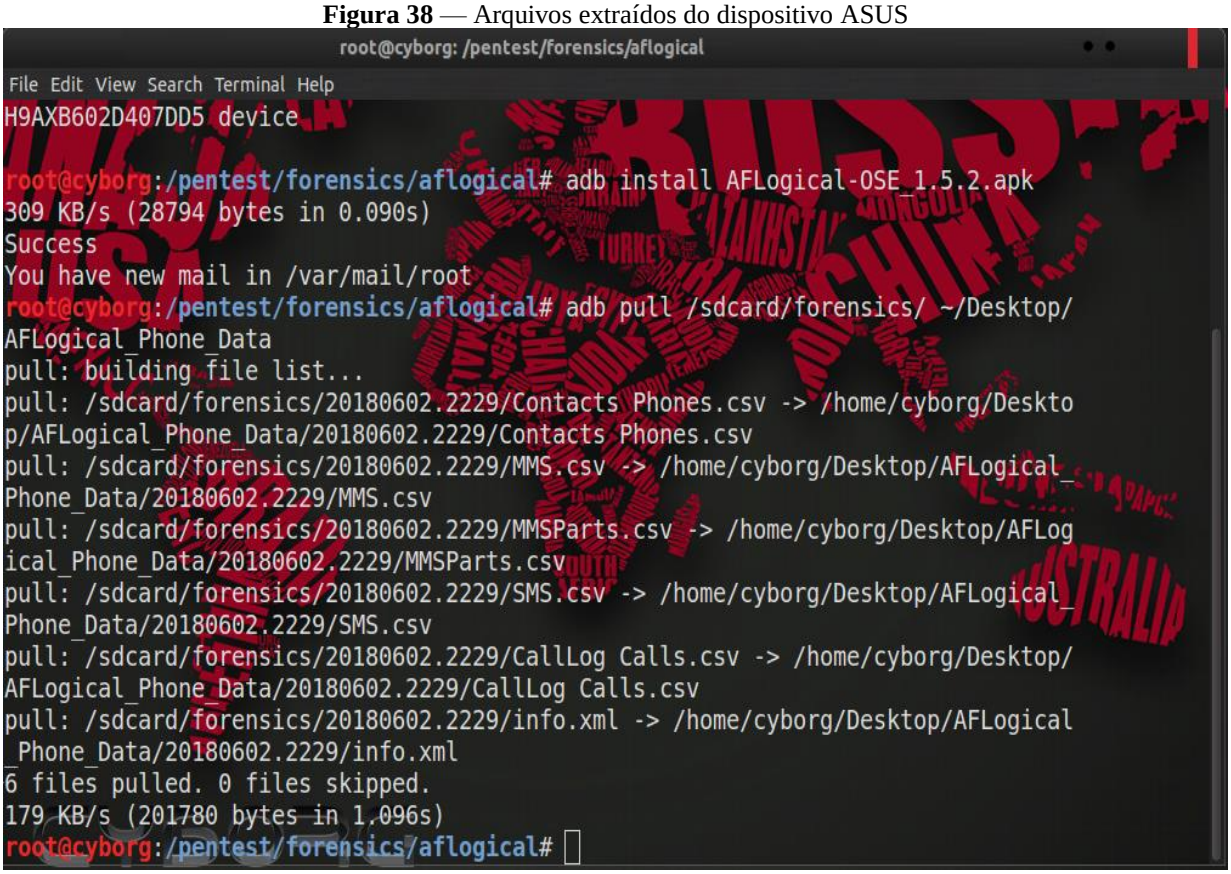
```
root@cyborg: mkdir ~/Desktop/AFLogical_Phone_Data
```

E agora realizaremos a extração dos dados, utilizando o seguinte comando:

```
root@cyborg:/pentest/forensics/aflogical# adb pull /sdcard/forensics/ ~/Desktop/AFLogical_Phone_Data
```

Em seguida aparecerá a quantidade de arquivos retirado e de arquivos que foram ignorados como mostra a imagem 38 abaixo.

Figura 38 — Arquivos extraídos do dispositivo ASUS



Fonte: Elaborado pelo próprio autor.

Os dados são armazenados em uma pasta nomeada com hora e data de aquisição, no diretório: `cyborg@cyborg:~/Desktop/AFLogical_Phone_Data$`

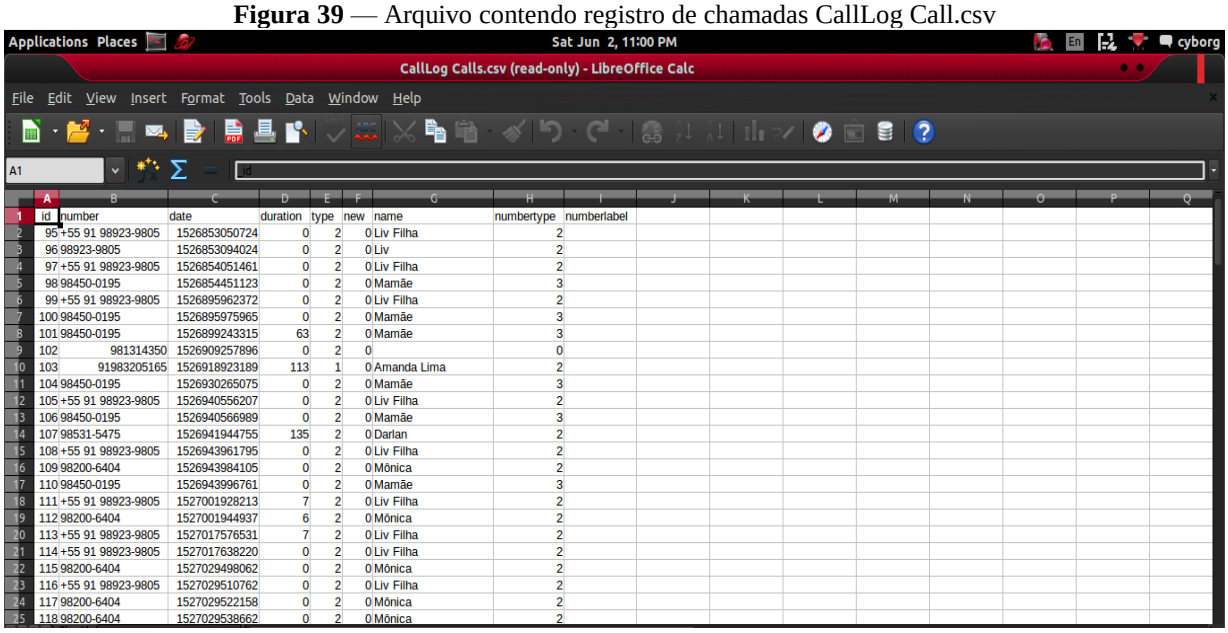
8.1.2 Exame dos arquivos obtidos via Aflogical_Ose

Após a extração dos dados através da aplicação "*AFLogical_OSE*", podemos visualizar dentro do diretório: `~/Desktop/AFLogical_Phone_Data$`, alguns arquivos armazenados em formato ".csv", que podem ser utilizados por um analista pericial. Estas informações, vão desde logs de chamas, registros de contatos, arquivo de informações xml e SMS.

Foi possível obter dado armazenados no arquivo CallLog Call.csv, contendo informação de registro de chamada. Este arquivo, contém informações como número de telefone, data de ligação, duração, tipo de chamada, nome das pessoas para qual foi feita a ligação. Estas

informações devem ser analisadas a critério do analista pericial, com o intuito de buscar e individualizar os contatos com os quais o proprietário do celular se comunicava.

Figura 39 — Arquivo contendo registro de chamadas CallLog Call.csv



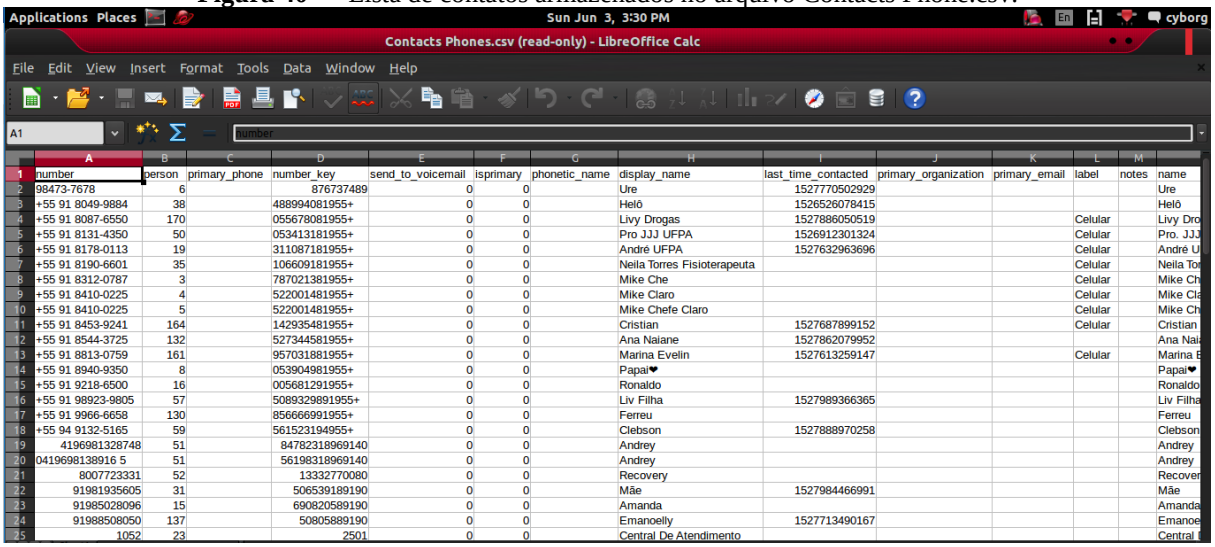
CallLog Calls.csv (read-only) - LibreOffice Calc

id	number	date	duration	type	new	name	numbertype	numberlabel
95	+55 91 98923-9805	1526853050724	0	2	0	Liv Filha	2	
96	98923-9805	1526853094024	0	2	0	Liv	2	
97	+55 91 98923-9805	1526854051461	0	2	0	Liv Filha	2	
98	98450-0195	1526854451123	0	2	0	Mamãe	3	
99	+55 91 98923-9805	1526895962372	0	2	0	Liv Filha	2	
100	98450-0195	1526895975965	0	2	0	Mamãe	3	
101	98450-0195	1526899243315	63	2	0	Mamãe	3	
102	981314350	1526909257896	0	2	0		0	
103	91983205165	1526918923189	113	1	0	Amanda Lima	2	
104	98450-0195	1526930265075	0	2	0	Mamãe	3	
105	+55 91 98923-9805	1526940556207	0	2	0	Liv Filha	2	
106	98450-0195	1526940566989	0	2	0	Mamãe	3	
107	98531-5475	1526941944755	135	2	0	Darlan	2	
108	+55 91 98923-9805	1526943961795	0	2	0	Liv Filha	2	
109	98200-6404	1526943984105	0	2	0	Mônica	2	
110	98450-0195	1526943996761	0	2	0	Mamãe	3	
111	+55 91 98923-9805	1527001928213	7	2	0	Liv Filha	2	
112	98200-6404	1527001944937	6	2	0	Mônica	2	
113	+55 91 98923-9805	1527017576531	7	2	0	Liv Filha	2	
114	+55 91 98923-9805	1527017638220	0	2	0	Liv Filha	2	
115	98200-6404	1527029498062	0	2	0	Mônica	2	
116	+55 91 98923-9805	1527029510762	0	2	0	Liv Filha	2	
117	98200-6404	1527029522158	0	2	0	Mônica	2	
118	98200-6404	1527029538662	0	2	0	Mônica	2	

Fonte: Elaborado pelo próprio autor.

Também foi possível obter no arquivo "*Contacts Phone.csv"*, como mostra a Figura 40 abaixo, uma lista com todos os nomes de contatos armazenados no smartphone, estas informações podem ser facilmente associadas a outros proprietários de linha telefônica.

Figura 40 — Lista de contatos armazenados no arquivo Contacts Phone.csv.



Contacts Phones.csv (read-only) - LibreOffice Calc

number	person	primary_phone	number_key	send_to_voicemail	isprimary	phonetic_name	display_name	last_time_contacted	primary_organization	primary_email	label	notes	name
98473-7678	6		876737489	0	0		Ure	1527770502929					Ure
+55 91 8049-9884	38		488994081955+	0	0		Helô	1526526078415					Helô
+55 91 8087-6550	170		055678081955+	0	0		Livy Drogas	1527886050519			Celular		Livy Dro
+55 91 8131-4350	50		053413181955+	0	0		Pro JJJ UFPA	1526912301324			Celular		Pro. JJJ
+55 91 8178-0113	19		311087181955+	0	0		André UFPA	1527632963696			Celular		André U
+55 91 8190-6601	35		106609181955+	0	0		Neila Torres Fisioterapeuta				Celular		Neila To
+55 91 8312-0787	3		787021381955+	0	0		Mike Che				Celular		Mike Ch
+55 91 8410-0225	4		522001481955+	0	0		Mike Claro				Celular		Mike Cl
+55 91 8410-0225	5		522001481955+	0	0		Mike Chefe Claro				Celular		Mike Ch
+55 91 8453-9241	164		142935481955+	0	0		Cristian	1527687899152			Celular		Cristian
+55 91 8544-3725	132		527344581955+	0	0		Ana Naiane	1527862079952					Ana Nai
+55 91 8813-0759	161		957031881955+	0	0		Marina Evelin	1527613259147			Celular		Marina E
+55 91 8940-9350	8		053904981955+	0	0		Papai❤						Papai❤
+55 91 9218-6500	16		005681291955+	0	0		Ronaldo						Ronaldo
+55 91 98923-9805	57		5089329891955+	0	0		Liv Filha	1527989366365					Liv Filha
+55 91 9966-6658	130		856666991955+	0	0		Ferreu						Ferreu
+55 94 9132-5165	59		561523194955+	0	0		Clebson	1527888970258					Clebson
4196981328748	51		84782318969140	0	0		Andrey						Andrey
04196981389165	51		56198318969140	0	0		Andrey						Andrey
8007723331	52		13332770080	0	0		Recovery						Recover
91981935605	31		506539189190	0	0		Mãe	1527984466991					Mãe
91985028096	15		690820589190	0	0		Amanda						Amanda
91988508050	137		50805889190	0	0		Emanoelly	1527713490167					Emanoe
1052	23		2501	0	0		Central De Atendimento						Central

Fonte: Elaborado pelo próprio autor.

8.1.3 Exames de Arquivos-Fotos obtidos via inspeção manual

Foi possível obter imagens extraídas do cartão de memória como mostra a Figura 41 logo abaixo.

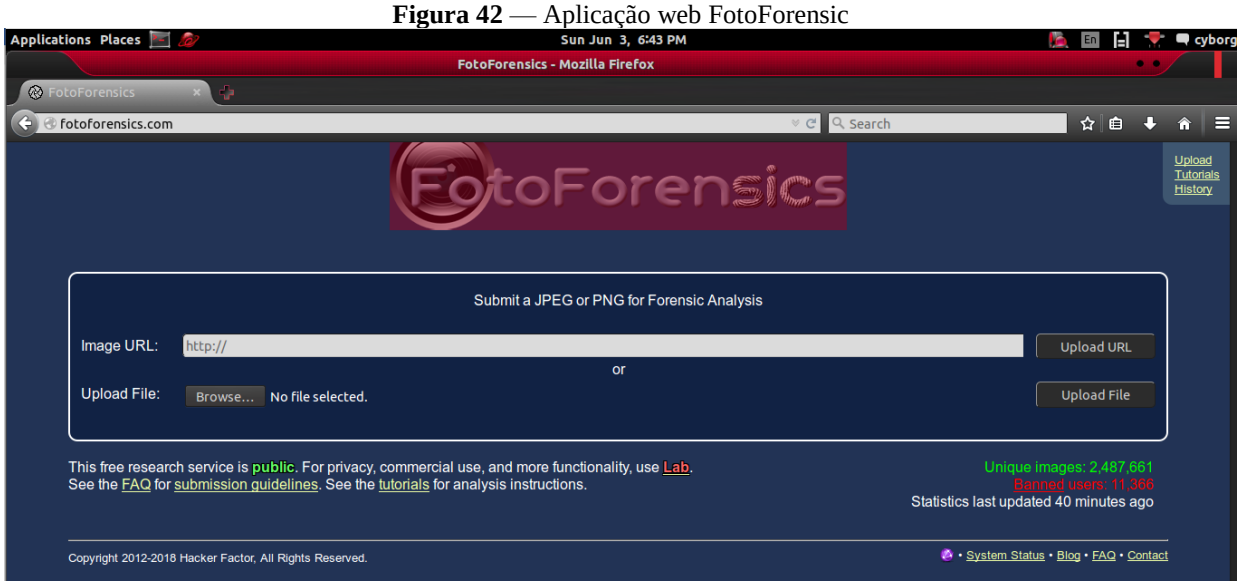
Figura 41 — Imagens extraído do cartão de memória

Fonte: Elaborado pelo próprio autor.

Afim de avaliar os metadados contidos em uma das fotos do dispositivo móvel analisado, foi utilizado o software "*FotoForensics*". Esta ferramenta possibilita a análise dos metadados contidos em uma imagem. Os metadados fornecem informações como, tipo de câmera usada para tirar a foto, localização, hora e data, dentre outras informações importantíssimas que podem ser utilizadas pelo Analista Pericial. Para análise de imagens digitais, devemos abrir o sistema "*FotoForensics*", via navegador web [27] (FOTOFORENSICS, 2018).

[27] Disponível para acesso no endereço: http://fotoforensics.com/

Figura 42 — Aplicação web FotoForensic



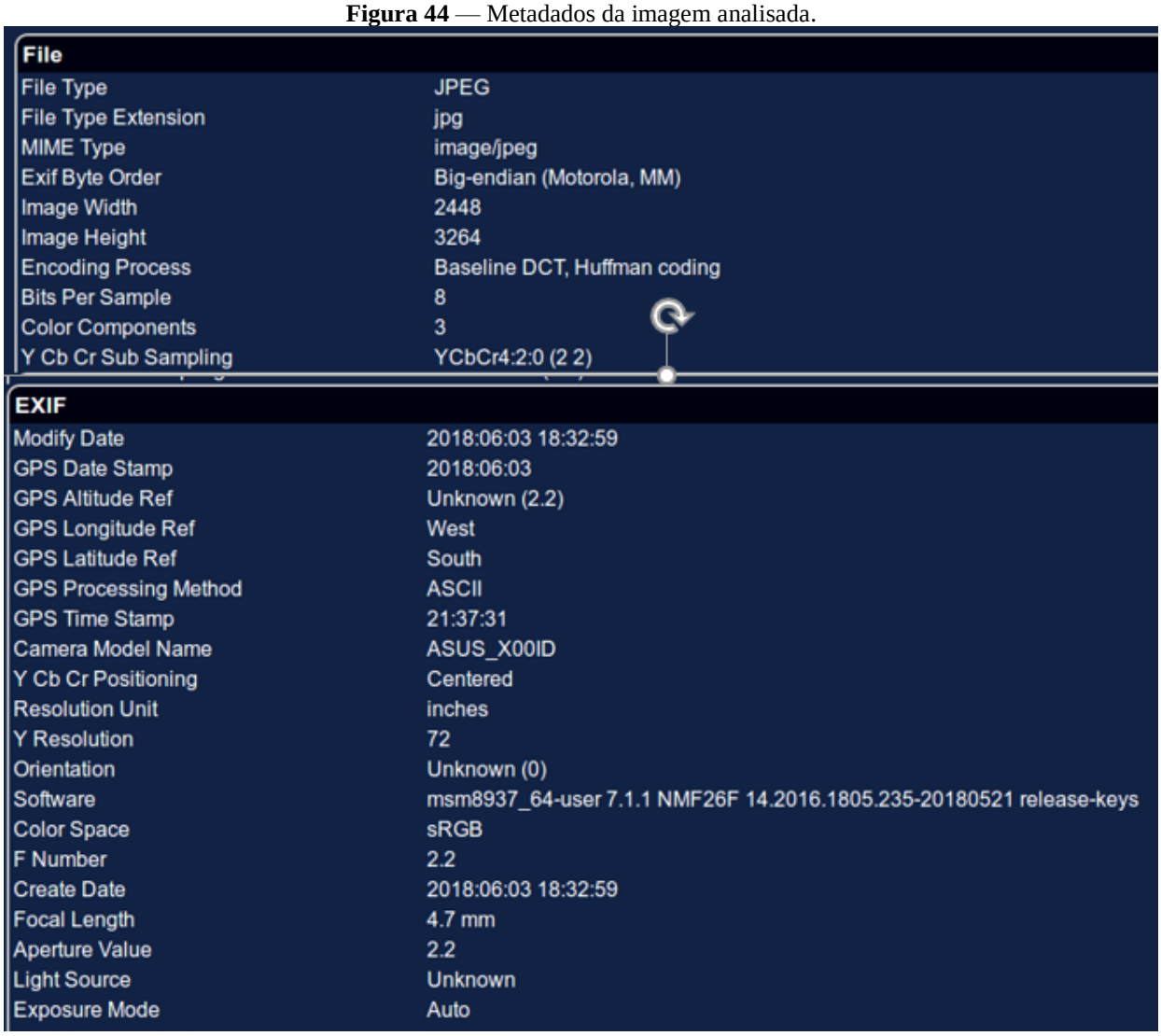
Fonte: Elaborado pelo próprio autor.

Figura 43 — Arquivo encontrado no cartão de memória do dispositivo móvel

Fonte: Elaborado pelo próprio autor.

Feito o carregamento das imagens podemos analisa-la, clicando na opção "*Metadata*" do menu "*Analysis"* disponível pela ferramenta "*fotoForenscis"*. Feito isso, podemos observado os dados coletados na Figura 43, logo abaixo.

Figura 44 — Metadados da imagem analisada.



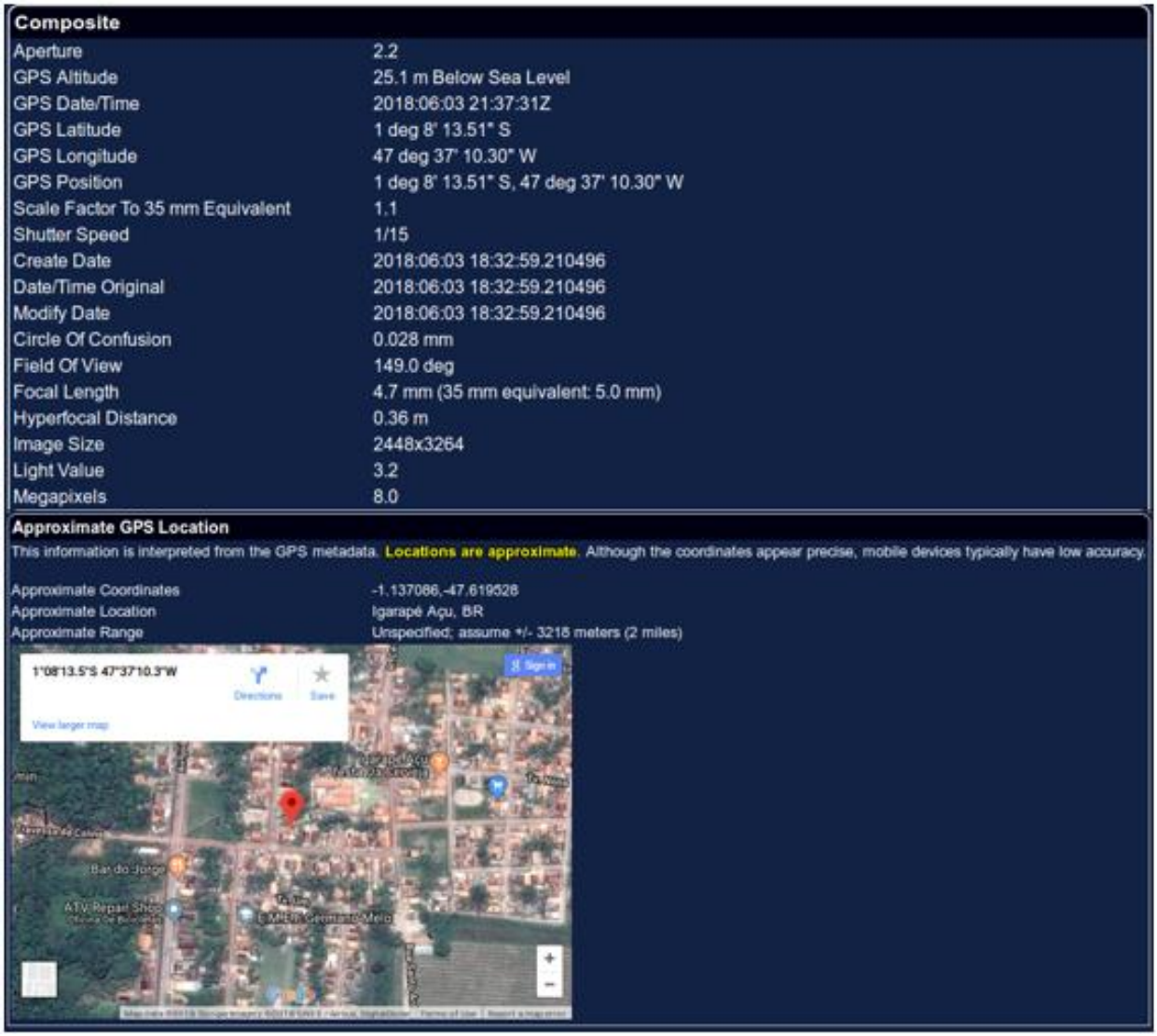
File	
File Type	JPEG
File Type Extension	jpg
MIME Type	image/jpeg
Exif Byte Order	Big-endian (Motorola, MM)
Image Width	2448
Image Height	3264
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
Y Cb Cr Sub Sampling	YCbCr4:2:0 (2 2)

EXIF	
Modify Date	2018:06:03 18:32:59
GPS Date Stamp	2018:06:03
GPS Altitude Ref	Unknown (2.2)
GPS Longitude Ref	West
GPS Latitude Ref	South
GPS Processing Method	ASCII
GPS Time Stamp	21:37:31
Camera Model Name	ASUS_X00ID
Y Cb Cr Positioning	Centered
Resolution Unit	inches
Y Resolution	72
Orientation	Unknown (0)
Software	msm8937_64-user 7.1.1 NMF26F 14.2016.1805.235-20180521 release-keys
Color Space	sRGB
F Number	2.2
Create Date	2018:06:03 18:32:59
Focal Length	4.7 mm
Aperture Value	2.2
Light Source	Unknown
Exposure Mode	Auto

Fonte: Elaborado pelo próprio autor.

Portanto, foi possível extrair utilizando a ferramenta "*FotoForenscis"*, dados importantes da imagem tais como: tipo de extensão da imagem, data hora de modificação, nome da câmera dentre outras informações. Além destas informações relevantes para a análise pericial é possível verificar a localização aproximada de onde a foto foi tirada atreveis da ferramenta, mas para que isso seja possível é necessário que o proprietário tenha tirado a foto, com os sistemas de localização do smartphone habilitado. Desta forma, foi possível definir de maneira satisfatório a localização de onde a imagem foi tirada, como mostra a figura 44

Figura 45 — Localização de onde a foto foi tirada



Fonte: Elaborado pelo próprio autor.

Muitos dispositivos móveis, incluindo smartphones e tablets, são capazes de capturar dados do sistema de posicionamento global (GPS) e armazená-los como metadados adicionais. As informações do GPS podem ajudar a identificar onde a foto foi tirada. (FOTOFORENSICS, 2018).

Partindo para uma análise mais aprofundada da fotografia encontra, tirada na data de 03:06:2018 as 21:37:31 (horário de Brasília), que possui em seus metadados as coordenadas geográficas: GPS Latitude 1 deg 8’ 13.51’ S, e GPS Longitude: 47 deg 7’ 10.30w, que aponta para uma região localizada na cidade de Igarapé-Açu/PA, nas proximidades do Bairro: Piçarreira.

8.2 Resultado 2: Aparelho ligado, sem bloqueio de tela padrão e com acesso a depuração USB ativada.

Este cenário tem como meta apresentar uma situação de exame de um *smartphone*, de um usuário intermediário, que utiliza aplicações mais complexas além daquelas fornecidas por padrão, fazendo uso de controle de depuração via USB, e controle de acesso padrão.

No momento do recolhimento do dispositivo o mesmo se encontra ligado, sendo automaticamente isolado da rede, através da ativação do modo avião. O modelo do *smartphone* deste cenário é o *Motorola, modelo Moto C Plus, Android 7.0*, ele permite a remoção do cartão SD para extração dos dados, o que facilitou a análise e gerenciamento do cartão de memória por meio de navegação manual e inspeção visual.

Como a Instalação do *Oracle VM VirtualBox Extension Pack* foi realizada anteriormente no estudo de caso RESULTADO 01, a estação forense já está habilitada para o reconhecimento do smartphone via UBS. O procedimento a ser realizado foi repetir o processo de ativação do dispositivo na máquina virtual, desta forma nós direcionamos até a máquina virtual "*CyborgForense*", clicamos em Dispositivo > USB, observe se está aparecendo o dispositivo conectado e de um clique no mesmo, feito isto abrirá a tela semelhante à Figura 26 e é necessário confirmar a conexão via USB no *smartphone*, como mostra a imagem 46 abaixo:

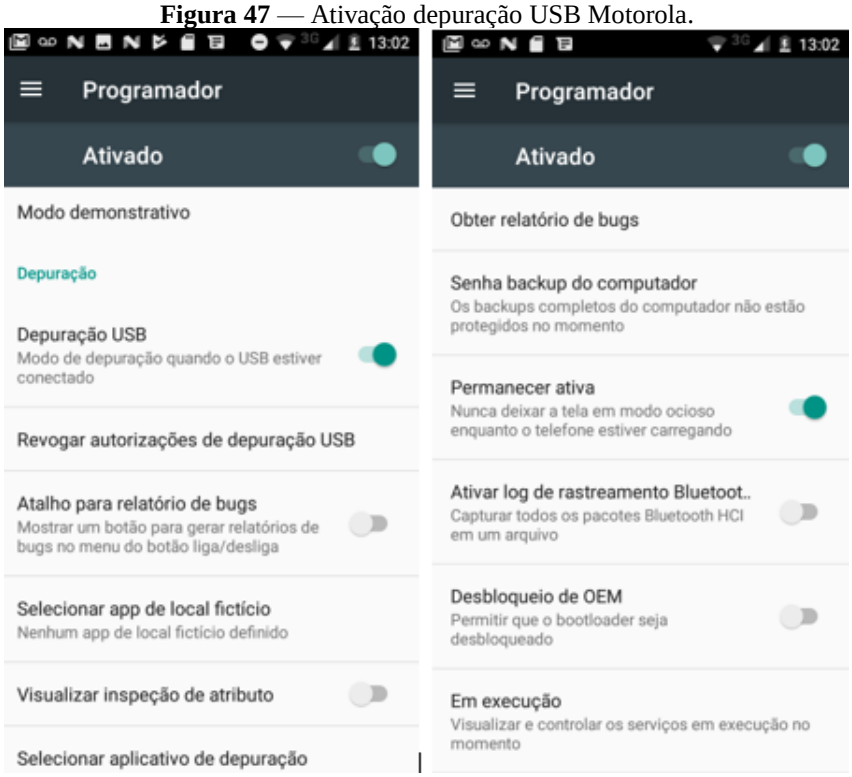
Figura 46 — Permissão de depuração de via USB

Fonte: Elaborado pelo próprio autor.

Da mesma forma como foi feito no RESULTADO 01, não será realizado uma tentativa de cópia *bit* a *bit* do cartão de memória, uma vez que já se observou de forma empírica que a máquina virtual "*CyborgForense"* não consegue reconhecer o dispositivo conectado em um adaptador de memória. Então, para a análise do cartão de memória, repetimos os testes utilizados no RESULTADO 01, que consistem em uma inspeção manual e visual por meio de navegação entre as pastas do dispositivo.

Após certificarmos que o aparelho foi reconectado via USB e que foi isolado de qualquer forma de comunicação posto em modo avião, e que as opções de "Depuração Via USB" e "Permanecer Ativo", estão habilitadas, o aparelho se encontra devidamente configurado para a análise, como mostra a figura 47 abaixo:

Figura 47 — Ativação depuração USB Motorola.



Fonte: Elaborado pelo próprio autor.

Após realizarmos as devidas configurações nos deslocamos até a ferramenta "*AFlogical_OSE*", nos sistemas "*CyborgForense"* e seguimos os passos:

`cyborg > Mobile Security > Devices Forensics > AFlogical Open Source Edition`,

Desta fora, como nos testes anteriores, a ferramentas abrirá em um terminal, solicitando para inserimos a senha de usuário root do sistema *CyborgForense* criado. Logo após inserirmos a senha digite os seguintes comandos:

```
root@cyborg:/pentest/forensics/aflogical# ls -l
```

Aparecerá a seguinte tela no terminal:

Figura 48 — Listando os dispositivos conectados

Fonte: Elaborado pelo próprio autor.

Para extrairmos os dados para o disposto devemos empurrar o "*AFLogical-OSE_1.5.2.apk*" para o dispositivo, mas, antes desse processo iremos conectar o dispositivo a estação forense, para isso foi utilizado o comando no terminal:

```
root@cyborg:/pentest/forensics/aflogical# adb devices
```

O resultado para a execução do comando:

Figura 49 — Executando adb devices para conectar ao dispositivo Motorola

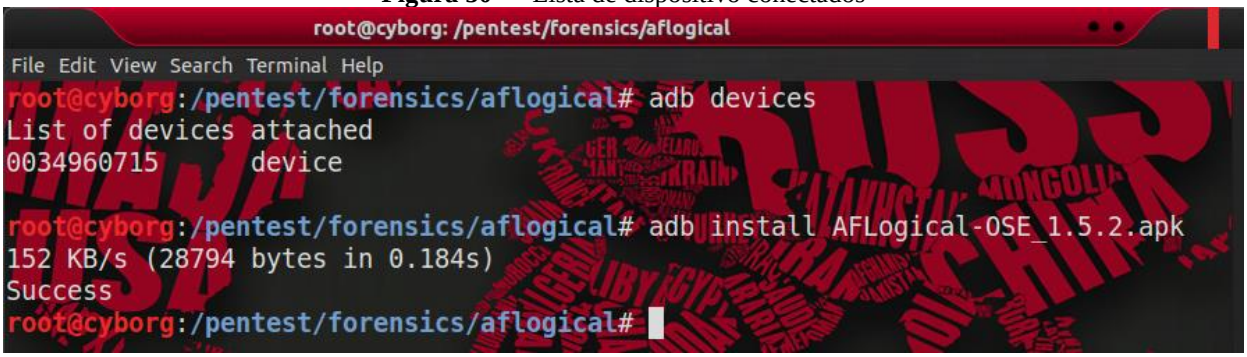
Fonte: Elaborado pelo próprio autor.

A seguir devemos "empurrar" o app para dentro do dispositivo executando o comando:

```
root@cyborg:/pentest/forensics/aflogical# adb install AFLogical-OSE_1.5.2.apk
```

Aparecerá a mensagem "***Success***" como mostra a figura 50 abaixo:

Figura 50 — Lista de dispositivo conectados

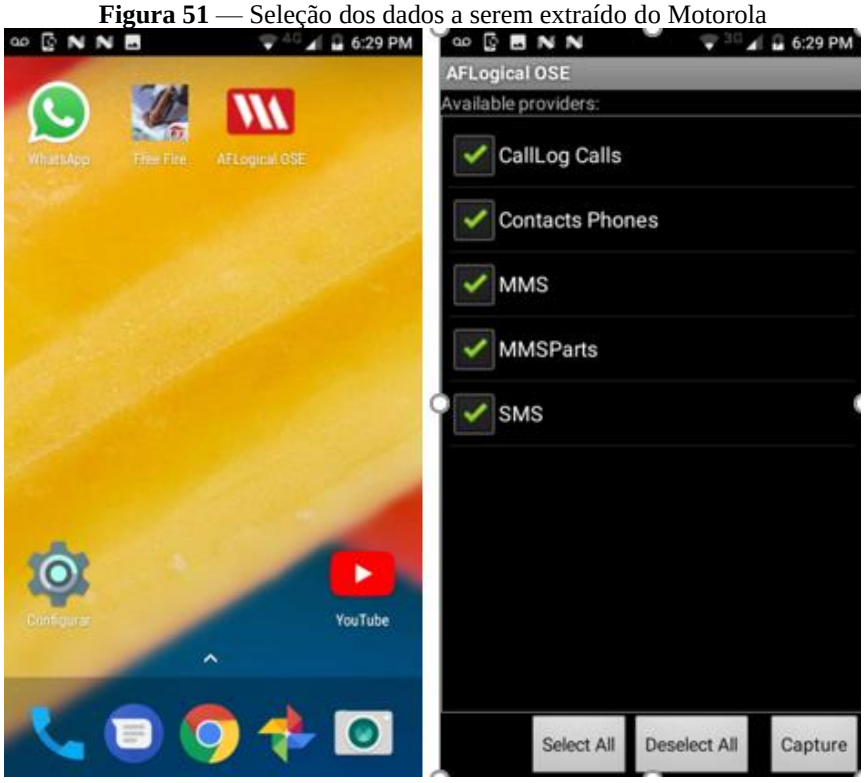


Fonte: Elaborado pelo próprio autor.

.

Esse APK foi instalado no dispositivo, então podemos abri-lo é escolher quais informações serão extraídas para o diretório `~/Desktop/AFLogical_Phone_Data`, que foi criado para o RESULTADO 01. A figura 51 mostra as opções disponíveis.

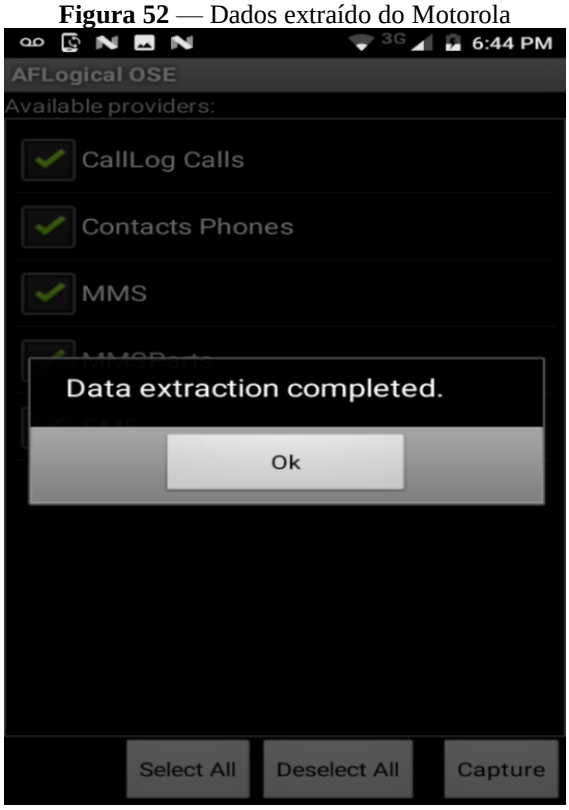
Figura 51 — Seleção dos dados a serem extraído do Motorola



Fonte: Elaborado pelo próprio autor.

Agora podemos puxar os dados extraído clicando no botão Capture da ferramenta, como mostra a imagem 52 abaixo:

Figura 52 — Dados extraído do Motorola

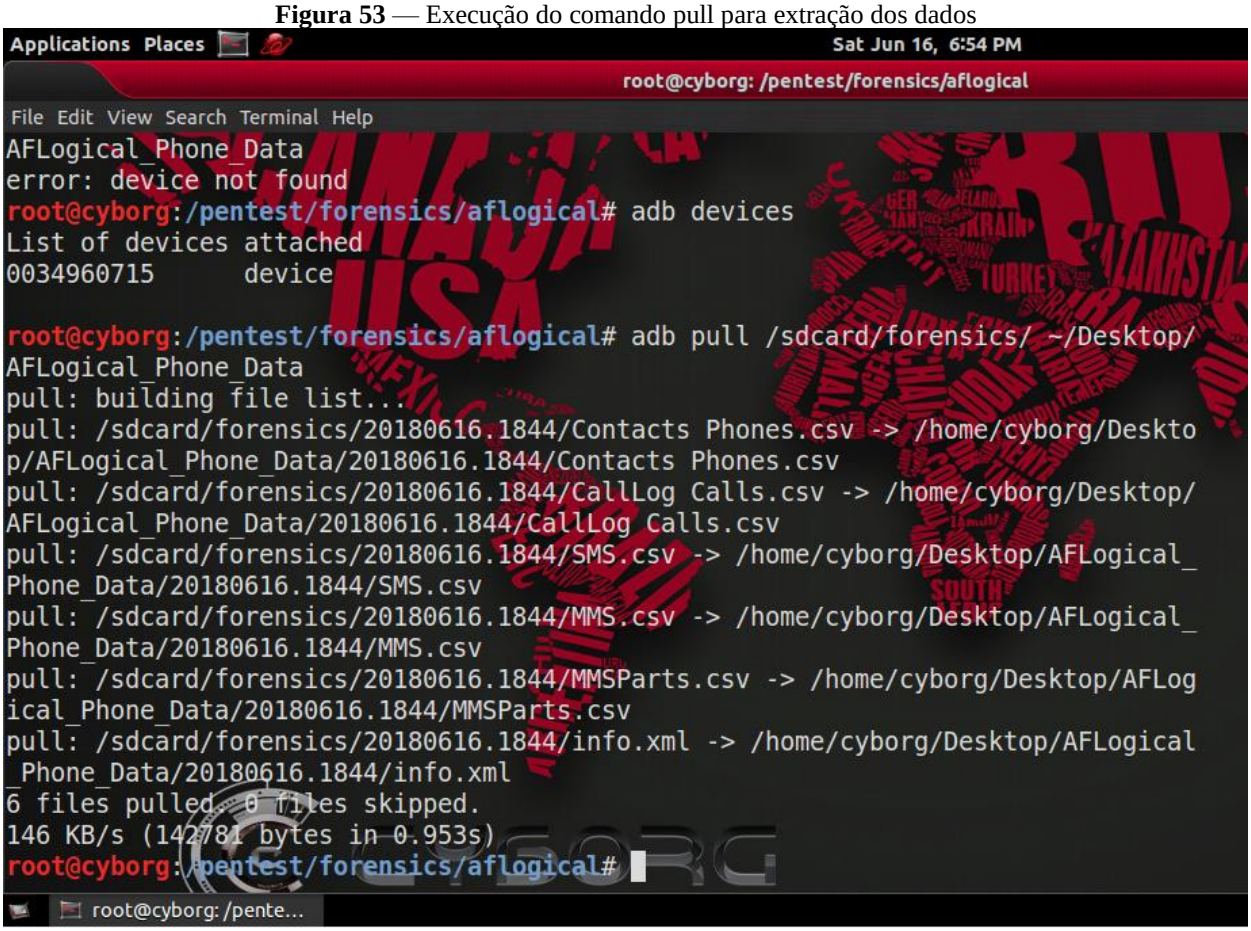


Fonte: Elaborado pelo próprio autor.

Em seguida, realizamos a extração dos dados para o diretório, executando o comando:

```
root@cyborg:/pentest/forensics/aflogical# adb pull /sdcard/forensics/
~/Desktop/AFLogical_Phone_Data
```

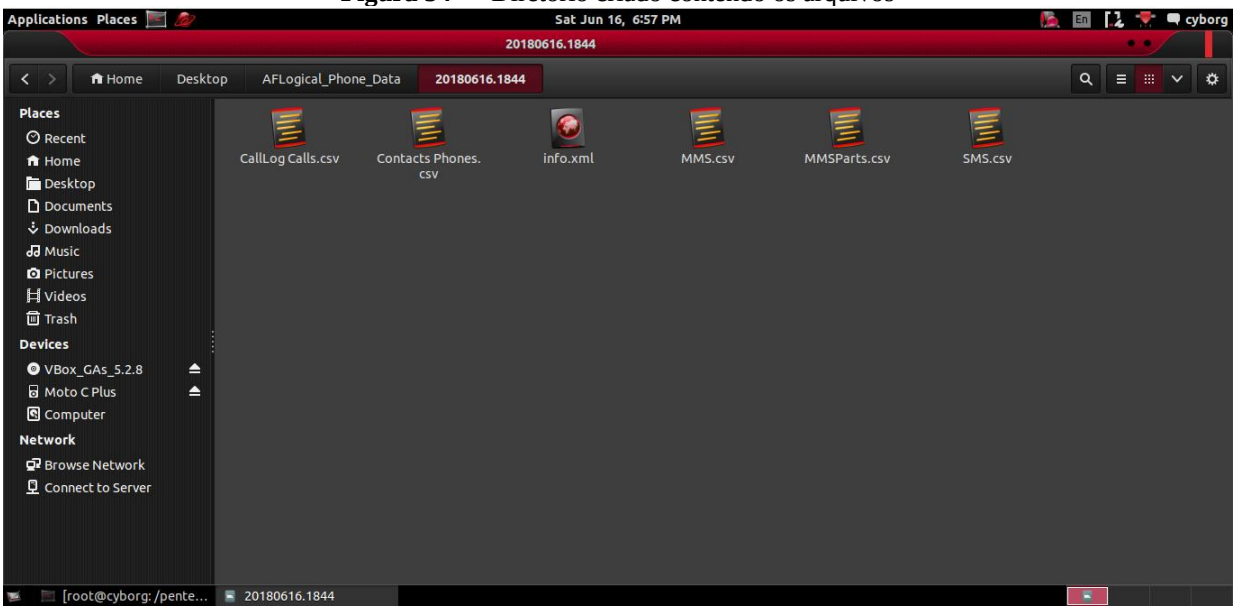
Figura 53 — Execução do comando pull para extração dos dados



Fonte: Elaborado pelo próprio autor.

A extração foi realizada como sucesso então podemos abrir o "*Nautilus"*. A nova pasta criada se encontra dentro do diretório e está nomeada como dada ano, dia e mês, como mostra figura 54 abaixo:

Figura 54 — Diretório criado contendo os arquivos



Fonte: Elaborado pelo próprio autor.

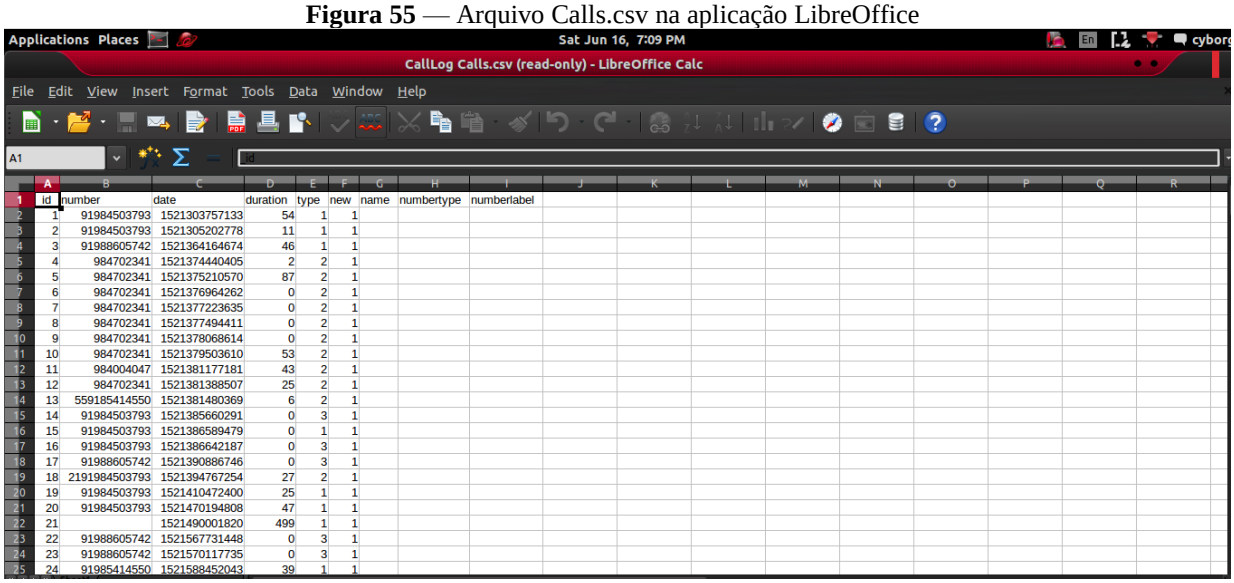
8.2.1 Exame dos arquivos via Aflogical-OSE_1.5.2

Após a extração dos dados obtidos através da aplicação AFLogical_OSE, podemos visualizar dentro do diretório, uma quantidade de arquivos relevante armazenados em arquivos do tipo *.csv*, que podem ser utilizados por um analista pericial. Estas informações, vão desde logs de chamas, registros de contatos, arquivo de informações xml e SMS.

Foi possível obter dado armazenados no arquivo CallLog Call.csv, contendo informação de registro de chamada. Estes arquivos, contém informações como número de telefone, data de ligação, duração, tipo de chamada.

Estas informações, devem ser analisadas a critério do analista pericial, com o intuito de buscar e determinar e individualizar pessoas com as quais o proprietário do celular mantinha contato. A figura 55 abaixo traz essas informações.

Figura 55 — Arquivo Calls.csv na aplicação LibreOffice



id	number	date	duration	type	new	name	numbertype	numberlabel
1	91984503793	1521303757133	54	1	1			
2	91984503793	1521305202778	11	1	1			
3	91988605742	1521364164674	46	1	1			
4	984702341	1521374440405	2	2	1			
5	984702341	1521375210570	87	2	1			
6	984702341	1521376964262	0	2	1			
7	984702341	1521377223635	0	2	1			
8	984702341	1521377494411	0	2	1			
9	984702341	1521378068614	0	2	1			
10	984702341	1521379503610	53	2	1			
11	984004047	1521381177181	43	2	1			
12	984702341	1521381388507	25	2	1			
13	559185414550	1521381480369	6	2	1			
14	91984503793	1521385660291	0	3	1			
15	91984503793	1521386589479	0	1	1			
16	91984503793	1521386642187	0	3	1			
17	91988605742	1521390886746	0	3	1			
18	2191984503793	1521394767254	27	2	1			
19	91984503793	1521410472400	25	1	1			
20	91984503793	1521470194808	47	1	1			
21		1521490001820	499	1	1			
22	91988605742	1521567731448	0	3	1			
23	91988605742	1521570117735	0	3	1			
24	91985414550	1521588452043	39	1	1			

Fonte: Elaborado pelo próprio autor.

Outros arquivos tais como: *ContactsPhone.csv*, *MMS.csv*, *MMSParts.csv*, *SMS.csv*, não continham tantas informações relevante que pudessem ser utilizadas.

8.2.2 Exames de arquivos (Fotos) obtidas via inspeção manual

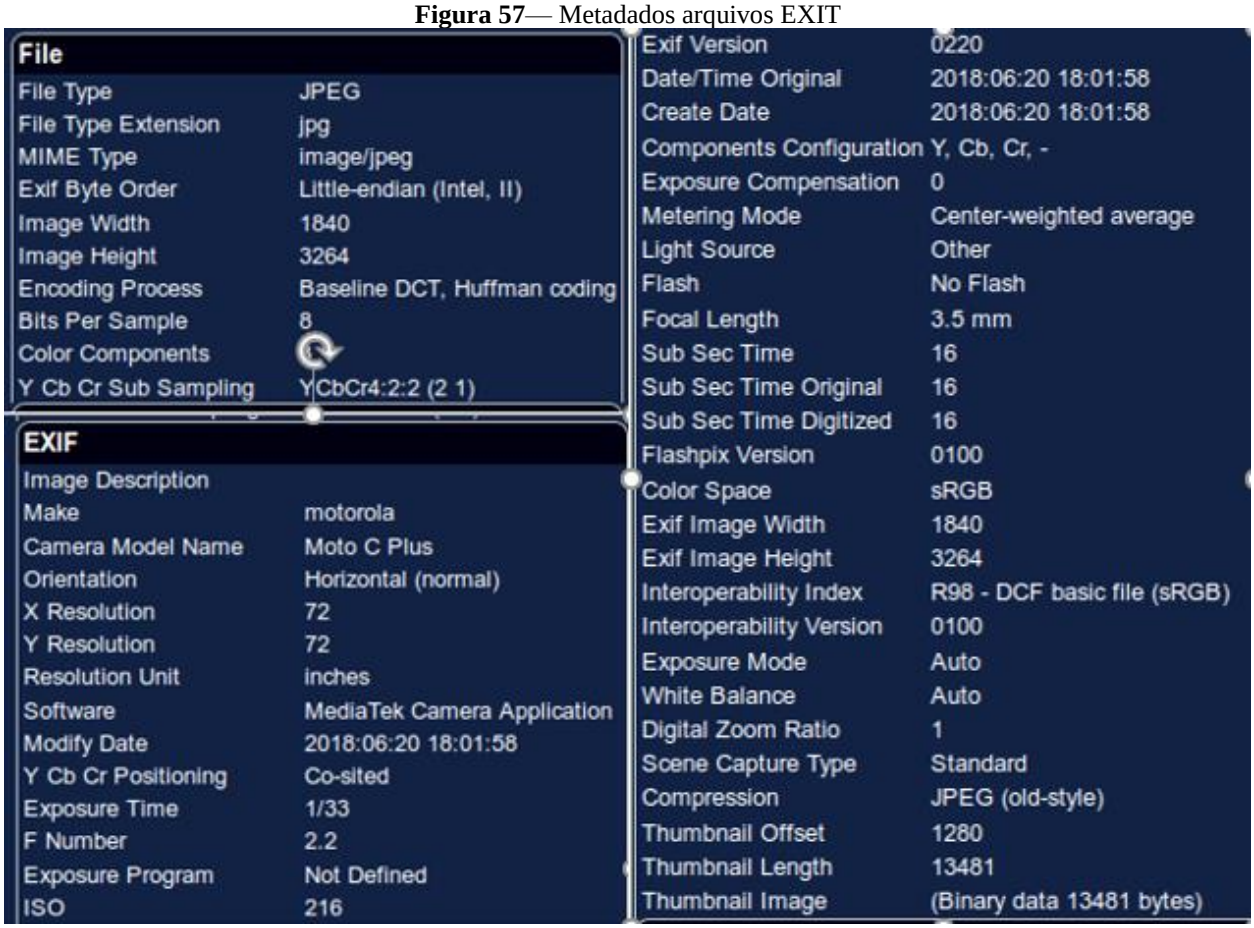
Além das informações que foram obtidas, podemos observar no RESULTADO 2 que no cartão de memória na pasta DCIM > Câmera, haviam 10 fotos de crianças. Afim de avaliar os metadados contidos em uma das fotos obtidas a critério do analista, foi utilizado o software "*FotoForensics"*. As informações obtidas com essa aplicação, como visto anteriormente, possibilitam a análise dos metadados contidos em uma imagem.

Figura 56 — Carregamento da imagem utilizando FotoForensics

Fonte: Elaborado pelo próprio autor.

Feito o carregamento da imagem, podemos analisa-la clicando na opção "*Metadata*" do menu *Analysis* disponível pela ferramenta "*FotoForenscis*". Feito isso, observamos os dados coletados na Figura 56, logo abaixo.

Figura 57— Metadados arquivos EXIT



Fonte: Elaborado pelo próprio autor.

No campo file dos metadados obtidos da imagem, foi possível observar claramente dados importantes como: Tipos de arquivo da imagem JPEG, dimensões da imagem, entre outras informações que podem ser usadas pelo analista pericial.

No campo EXIF (Exchangeable Image File), é normalmente utilizado pelo fabricante das câmeras para identificar informações sobre as câmeras utilizadas na foto, incluindo informações como *timestamps***,** nome do modelo da câmera, *Modity Date*, que inclui: data e horas em que foi tirada a imagem, e muitas outras informações importantes.

Não foi possível observar informações de localização de onde foi tirada a imagem, uma vez que, estas informações necessitam que o GPS do dispositivo esteja ligado, como foi mostrado na Figura 56.

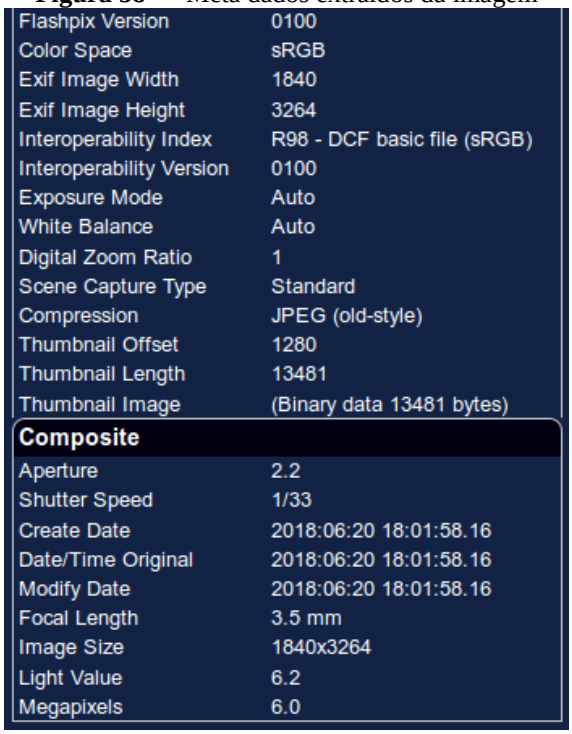
Também foi possível encontra na memória interna do dispositivo na pasta *thumnails* um total de 50 imagens no formato JPG.

Também foi possível visualizar navegar pela memória interna compartilhada > *Whatsap* > *Midia*. Estas pastas podem conter diversas informações importante como imagens, textos, vídeos e áudios, e muitas outras informações relevantes.

Por fim, foi possível definir por meio de navegação manual no dispositivo analisado a conta de e-mail utilizada para no dispositivo, da qual: todosgostamdocris926@gmail.com.

A definição da conta utilizada no dispositivo é muito importante, já que ele está associa a quase todas as aplicações que possam vir a ser utilizadas pelo usuário.

Figura 58 — Meta dados extraídos da imagem



Flashpix Version	0100
Color Space	sRGB
Exif Image Width	1840
Exif Image Height	3264
Interoperability Index	R98 - DCF basic file (sRGB)
Interoperability Version	0100
Exposure Mode	Auto
White Balance	Auto
Digital Zoom Ratio	1
Scene Capture Type	Standard
Compression	JPEG (old-style)
Thumbnail Offset	1280
Thumbnail Length	13481
Thumbnail Image	(Binary data 13481 bytes)
Composite	
Aperture	2.2
Shutter Speed	1/33
Create Date	2018:06:20 18:01:58.16
Date/Time Original	2018:06:20 18:01:58.16
Modify Date	2018:06:20 18:01:58.16
Focal Length	3.5 mm
Image Size	1840x3264
Light Value	6.2
Megapixels	6.0

Fonte: Elaborado pelo próprio autor.

8.3 Resultado 3: Aparelho ligado, com bloqueio de tela padrão e com acesso a depuração UBS ativada.

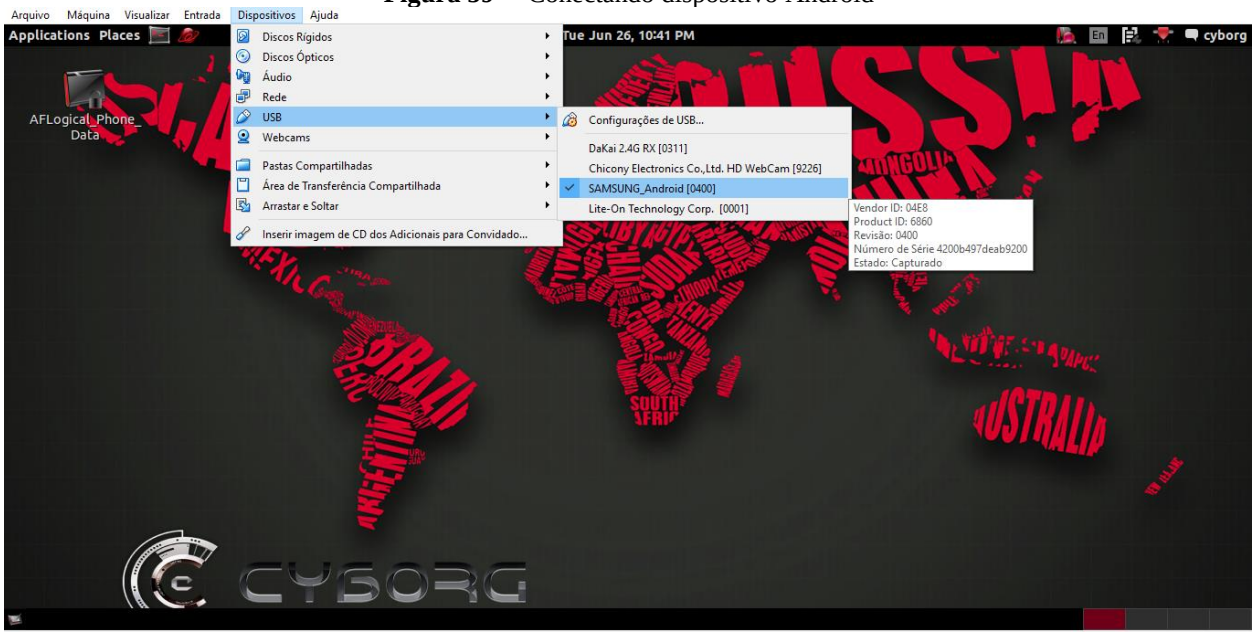
Este teste tem como objetivo apresentar a análise de um smartphone de um usuário expert, que utiliza bem os recursos de hardware e software do dispositivo, que possui conhecimento para instalação e customização do sistema, controle de acesso ativo, acesso a depuração (ADB) ativo e com permissões de super usuário instalada.

No momento do recolhimento do dispositivo o mesmo se encontra ligado, sendo automaticamente isolado da rede, através da ativação do modo avião. O modelo do smartphone deste cenário é o *Samsung Galaxy J1 (2016), Modelo SM – J120H***.** Ele permite a remoção do cartão

SD para extração dos dados, o que facilitou a análise e gerenciamento do cartão de memória por meio de navegação manual e inspeção visual.

A instalação do *Oracle VM VirtualBox Extension Pack* foi realizada anteriormente no tópico 11, desta forma a estação forense já está habilitada para o reconhecimento, do *smartphone* via UBS o procedimento a ser realizado foi repetir o processo de ativação do dispositivo na máquina virtual, desta forma nós direcionamos até a máquina virtual "*CyborgForense*", clicamos em Dispositivo > USB, observe se está aparecendo o dispositivo conectado e de um clique no mesmo, como ilustra a figura 58 abaixo.

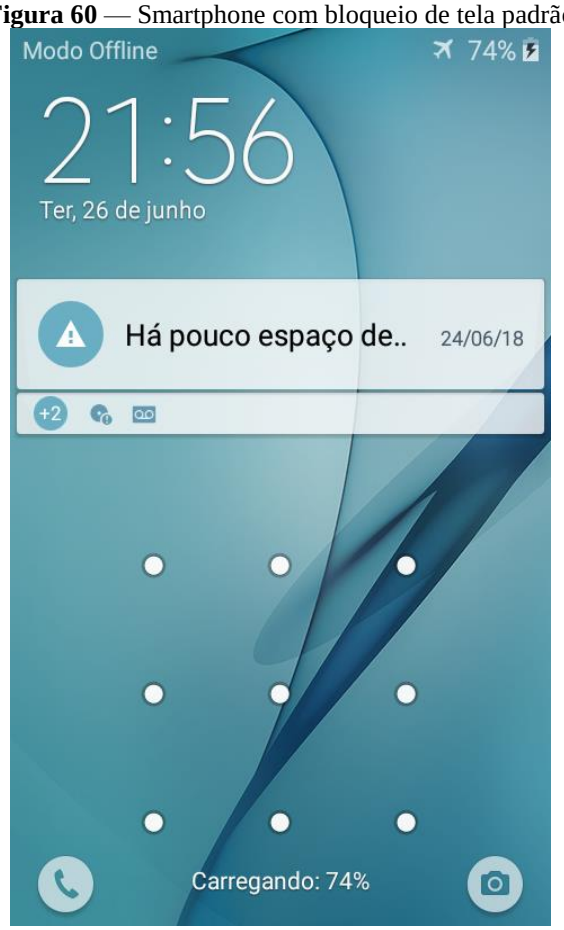
Figura 59— Conectando dispositivo Android



Fonte: Elaborado pelo próprio autor.

Com o celular devidamente conectado, ligado, sendo isolado das redes posto em modo avião e não sendo possível desbloqueá-lo, como mostra a imagem de bloqueio padrão abaixo.

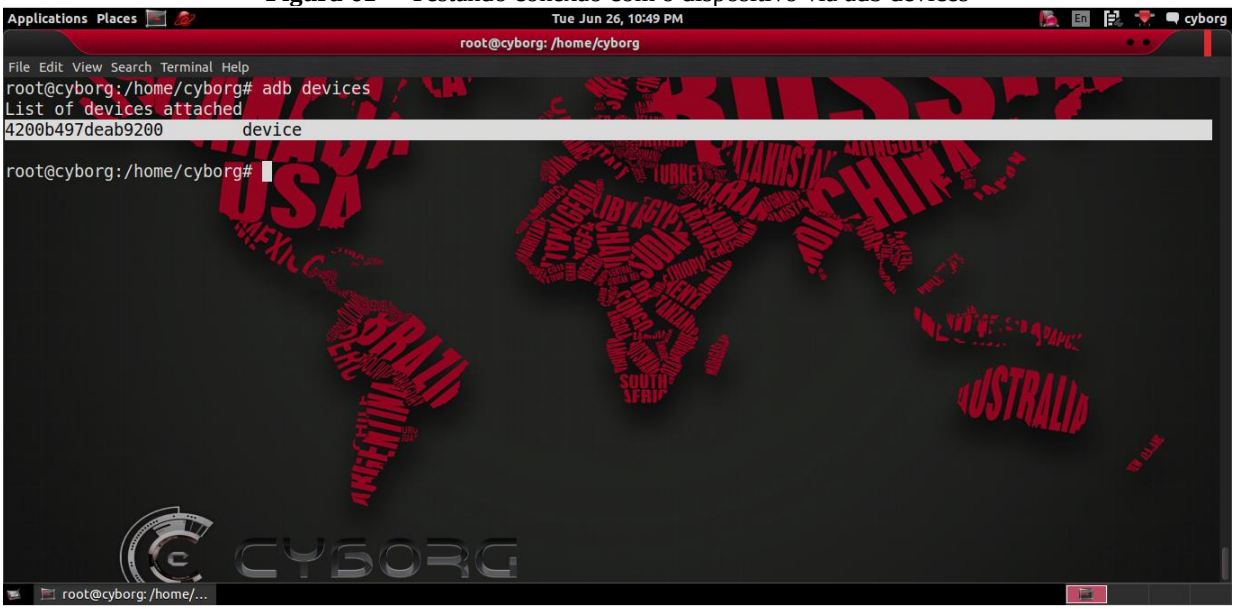
Figura 60 — Smartphone com bloqueio de tela padrão



Fonte: Elaborado pelo próprio autor.

Mesmo o smartphone possuindo bloqueio de tela padrão, foi possível realizar uma conexão de depuração UBS, com sucesso, da estação pericial "*CyborgForense"* com o smartphone analisado, como mostra a figura 60 logo abaixo.

Figura 61— Testando conexão com o dispositivo via adb devices



Fonte: Elaborado pelo próprio autor.

Assim como no RESULTADO 01 e RESULTADO 02, não será realizado uma tentativa de cópia bit a bit do cartão de memorial, com já se observou de forma empiria que a máquina virtual "*CyborgForense"* não consegue reconhecer o dispositivo conectado em um adaptador de memória. Então, para a análise do cartão de memória, repetimos os testes utilizados no RESULTADO 01 e RESULTADO 02, que consistem em uma inspeção manual e visual por meio de navegação entre as pastas do dispositivo.

Como sabe-se que o dispositivo analisado, foi reconhecido pela estação forense, como mostra o resultado na Figura 60, podemos inferir que o dispositivo já está com a opção de depuração via USB ativada, desta maneira prosseguimos com a instalação da ferramenta *AFLogical-OSE*.

8.3.1 Instalação da ferramenta Aflogical-Ose nos Smartphone Samsung J1

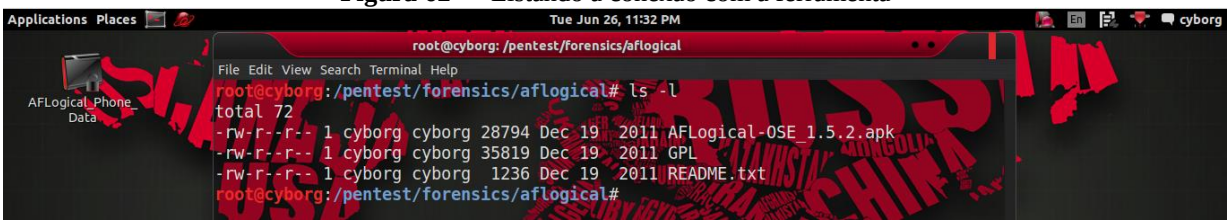
Neste momento, nos deslocamos até a ferramenta AFlogical-OSE nos sistemas "*CyborgForense"* e seguimos os passos: *cyborg* > *Mobile Security* > *Devices Forensics* > *AFlogical Open Source Edition*, assim como nos testes anteriores, a ferramentas abrirá em um terminal, solicitando para inserimos a senha de usuário root do sistema "*CyborgForense"*.

Logo após inserirmos a senha digite os seguintes comandos:

root@cyborg:/pentest/forensics/aflogical# ls -l

Aparecerá a seguinte mensagem no terminal:

Figura 62 — Listando a conexão com a ferramenta



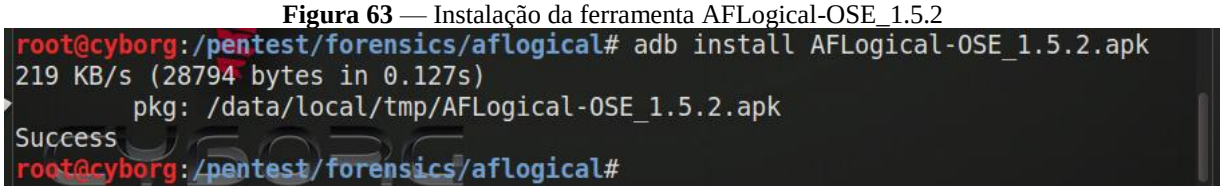
Fonte: Elaborado pelo próprio autor.

Em seguida "empurramos" o *App* para dentro do dispositivo executando o comando:

`root@cyborg:/pentest/forensics/aflogical# adb install AFLogical-OSE_1.5.2.apk`

Aparecerá a mensagem "***Success***" como mostra a figura 62 abaixo:

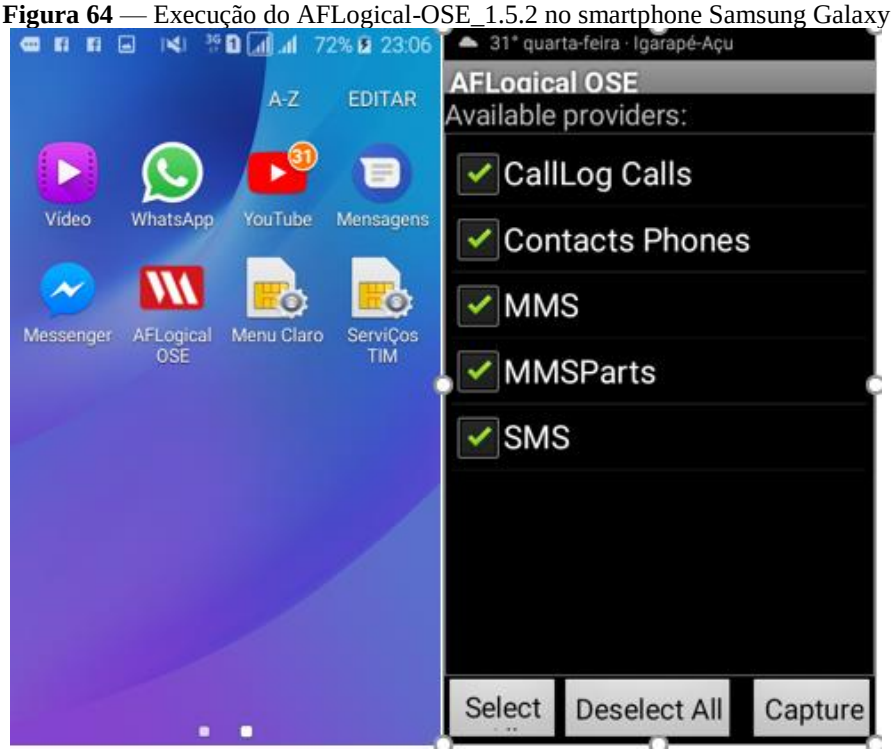
Figura 63 — Instalação da ferramenta AFLogical-OSE_1.5.2



Fonte: Elaborado pelo próprio autor.

Feito isso, direciona-se até a ferramenta *AFLogical-OSE* instalada no dispositivo, executamos a ferramenta e escolhemos todas opções para captura e extração de informações, como mostra a Figura 63, logo abaixo:

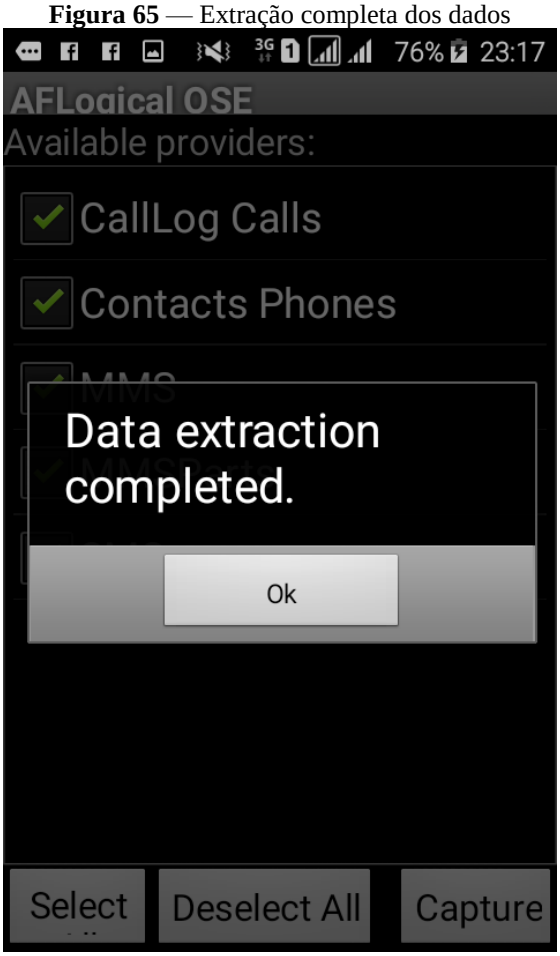
Figura 64 — Execução do AFLogical-OSE_1.5.2 no smartphone Samsung Galaxy



Fonte: Elaborado pelo próprio autor.

Agora podemos puxar os dados extraído clicando no botão Capture da ferramenta, como mostra a imagem abaixo:

Figura 65 — Extração completa dos dados



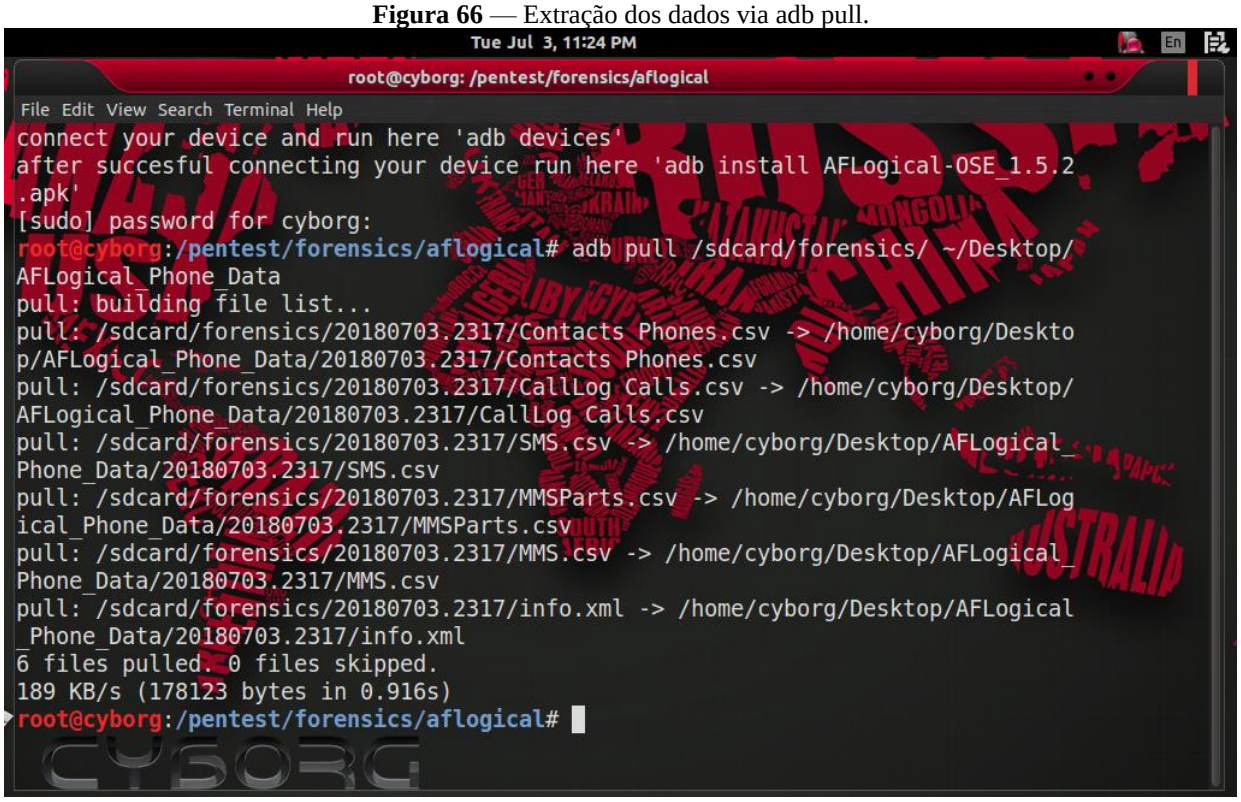
Fonte: Elaborado pelo próprio autor.

Em seguida, realizamos a extração dos dados para o diretório executando o comando:

```
root@cyborg:/pentest/forensics/aflogical# adb pull /sdcard/forensics/ ~/Desktop/AFLogical_Phone_Data
```

Feito isso, aparecerá no terminal a quantidade de arquivos extraídos e ignorados na tela do terminal, como monstra a imagem 64 abaixo.

Figura 66 — Extração dos dados via adb pull.



Fonte: Elaborado pelo próprio autor.

.

Realizada a extração, podemos abrir o "*Nautilus"* e verificar que a nova pasta com os dados foi criada dentro do diretório: `home/cyborg/Desktop/AFLogical_Phone_Data`, como mostra a Figura 65 logo abaixo:

Figura 67 — Pasta 20180703.2317 com os dados extraídos

Fonte: Elaborado pelo próprio autor.

8.3.2 Exames dos dados extraídos via Aflogical-OSE

Depois da extração dos dados obtidos através da aplicação *AFLogical_OSE_1.2.5*, foi possível visualizar dentro do diretório: `/home/cyborg/Desktop/AFLogical_Phone_Data`, uma quantidade de dados armazenados em uma pasta com data e hora de aquisição.

As informações armazenadas nesta pasta são do tipo ".csv". Estas informações, vão desde logs de chamas, registros de contatos, arquivo de informações xml e SMS.

Foi possível obter dado armazenados no arquivo "*CallLog Call.csv*", contendo informação de registro de chamada. Estes arquivos, contém informações como número de telefone, data de ligação, duração, tipo de chamada.

Estas informações, devem ser analisadas a critério do analista pericial, com o intuito de buscar e determinar e individualizar pessoas com as quais o proprietário do celular mantinha contato. A figura 66 abaixo traz essas informações:

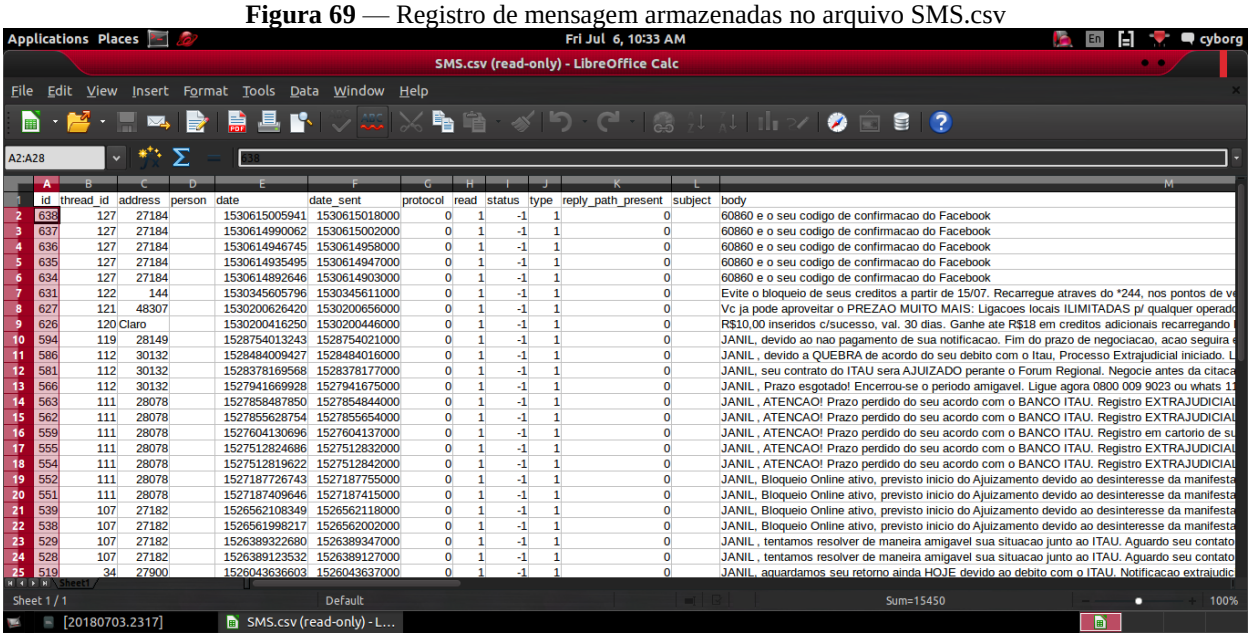
Figura 68 — Registro de chamadas do arquivo Calllog Call.csv

	A	B	C	D	E	F	G	H	I
1	id	number	date	duration	type	new	name	numbertype	numberlabel
2	2597	985358896	1530655006892	0	2	0			
3	2596	985358896	1530654980212	0	2	0			
4	2595	985163901	1530654917383	0	2	0	Luzia Claro	2	SIM
5	2594	985163901	1530654902550	0	2	0	Luzia Claro	2	SIM
6	2593	985163901	1530654855820	0	2	0	Luzia Claro	2	SIM
7	2592	985163901	1530654839864	0	2	0	Luzia Claro	2	SIM
8	2591	985163901	1530654811625	0	2	0	Luzia Claro	2	SIM
9	2585	985163901	1530568374244	60	2	0	Luzia Claro	2	SIM
10	2578	981981088	1530376419432	70	2	0	Malinhe Irmã	2	
11	2577	30883820	1530363539648	157	2	0	Guilherme Otic	2	SIM
12	2576	91984632694	1530353255653	77	1	0	Janinha Claro	2	
13	2569	91984632694	1530276854827	1098	1	0	Janinha Claro	2	
14	2567	91984503793	1530276547794	0	3	0	Marcela Claro	2	
15	2565	984597064	1530224211798	0	2	0	Jamilly Claro	2	
16	2563	985456687	1530223776394	40	2	0	Jean Claro	2	
17	2562	984734311	1530219695118	86	1	0	Jessica	2	SIM
18	2553	30883820	1529934516262	0	2	0	Guilherme Otic	2	SIM
19	2550	985456687	1529929326744	78	2	0	Jean Claro	2	
20	2549	91988950360	1529923764612	121	1	0	Josy Sodre	2	
21	2548	96981389165	1529868776653	1702	1	0	Andrey Macapa	2	
22	2544	37213498	1529764753616	89	1	0	CEOPCastanhal	2	
23	2543	37213498	1529763998131	182	1	0	CEOPCastanhal	2	
24	2510	96981389165	1529250853475	2182	1	0	Andrey Macapa	2	
25	2509	91984503793	1529232560131	114	1	0	Marcela Claro	2	

Fonte: Elaborado pelo próprio autor.

No mesmo diretório: /home/cyborg/Desktop/AFLogical_Phone_Data/20180703.2317, foi possível observar no arquivo SMS.csv, o registro de mensagens trocadas pelo usuário. O registro de mensagem pode ser visualizado na imagem 67 logo abaixo.

Figura 69 — Registro de mensagem armazenadas no arquivo SMS.csv



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id	thread_id	address	person	date	date_sent	protocol	read	status	type	reply_path_present	subject	body
2	638	127	27184		1530615005941	1530615018000	0	1	-1	1	0		60860 e o seu codigo de confirmacao do Facebook
3	637	127	27184		1530614990062	1530615002000	0	1	-1	1	0		60860 e o seu codigo de confirmacao do Facebook
4	636	127	27184		1530614946745	1530614958000	0	1	-1	1	0		60860 e o seu codigo de confirmacao do Facebook
5	635	127	27184		1530614935495	1530614947000	0	1	-1	1	0		60860 e o seu codigo de confirmacao do Facebook
6	634	127	27184		1530614892646	1530614903000	0	1	-1	1	0		60860 e o seu codigo de confirmacao do Facebook
7	631	122	144		1530345605796	1530345611000	0	1	-1	1	0		Evite o bloqueio de seus creditos a partir de 15/07. Recarregue atraves do *244, nos pontos de v
8	627	121	48307		1530200626420	1530200656000	0	1	-1	1	0		Vc ja pode aproveitar o PREZAO MUITO MAIS: Ligacoes locais ILIMITADAS p/ qualquer operad
9	626	120	Claro		1530200416250	1530200446000	0	1	-1	1	0		R$10,00 inseridos c/sucesso, val. 30 dias. Ganhe ate R$18 em creditos adicionais recarregando
10	594	119	28149		1528754013243	1528754021000	0	1	-1	1	0		JANIL, devido ao nao pagamento de sua notificacao. Fim do prazo de negociacao, acao seguira
11	586	112	30132		1528484009427	1528484016000	0	1	-1	1	0		JANIL , devido a QUEBRA de acordo do seu debito com o Itau, Processo Extrajudicial iniciado. L
12	581	112	30132		1528378169568	1528378177000	0	1	-1	1	0		JANIL, seu contrato do ITAU sera AJUIZADO perante o Forum Regional. Negocie antes da citaca
13	566	112	30132		1527941669928	1527941675000	0	1	-1	1	0		JANIL , Prazo esgotado! Encerrou-se o periodo amigavel. Ligue agora 0800 009 9023 ou whats 1
14	563	111	28078		1527858487850	1527854844000	0	1	-1	1	0		JANIL , ATENCAO! Prazo perdido do seu acordo com o BANCO ITAU. Registro EXTRAJUDICIAL
15	562	111	28078		1527855628754	1527855654000	0	1	-1	1	0		JANIL , ATENCAO! Prazo perdido do seu acordo com o BANCO ITAU. Registro EXTRAJUDICIAL
16	559	111	28078		1527604130696	1527604137000	0	1	-1	1	0		JANIL , ATENCAO! Prazo perdido do seu acordo com o BANCO ITAU. Registro em cartorio de su
17	555	111	28078		1527512824686	1527512832000	0	1	-1	1	0		JANIL , ATENCAO! Prazo perdido do seu acordo com o BANCO ITAU. Registro EXTRAJUDICIAL
18	554	111	28078		1527512819622	1527512842000	0	1	-1	1	0		JANIL , ATENCAO! Prazo perdido do seu acordo com o BANCO ITAU. Registro EXTRAJUDICIAL
19	552	111	28078		1527187726743	1527187755000	0	1	-1	1	0		JANIL, Bloqueio Online ativo, previsto inicio do Ajuizamento devido ao desinteresse da manifesta
20	551	111	28078		1527187409646	1527187415000	0	1	-1	1	0		JANIL, Bloqueio Online ativo, previsto inicio do Ajuizamento devido ao desinteresse da manifesta
21	539	107	27182		1526562108349	1526562118000	0	1	-1	1	0		JANIL, Bloqueio Online ativo, previsto inicio do Ajuizamento devido ao desinteresse da manifesta
22	538	107	27182		1526561998217	1526562002000	0	1	-1	1	0		JANIL, Bloqueio Online ativo, previsto inicio do Ajuizamento devido ao desinteresse da manifesta
23	529	107	27182		1526389322680	1526389347000	0	1	-1	1	0		JANIL , tentamos resolver de maneira amigavel sua situacao junto ao ITAU. Aguardo seu contato
24	528	107	27182		1526389123532	1526389127000	0	1	-1	1	0		JANIL , tentamos resolver de maneira amigavel sua situacao junto ao ITAU. Aguardo seu contato
25	519	34	27900		1526043636603	1526043637000	0	1	-1	1	0		JANIL, aguardamos seu retorno ainda HOJE devido ao debito com o ITAU. Notificacao extrajudic

Fonte: Elaborado pelo próprio autor.

Por fim, não foi possível categorizar nenhuma informação contido nos outros arquivos que foram criados resultantes da extração das informações via AFLogical-OSE.

8.3.3 Extração Manual

Como não foi possível criar uma imagem do cartão de memória do dispositivo para fins de análise, realizamos uma extração manual acessando o sistema de arquivos via "*Nautilus"*, possibilitando a cópia de arquivos ou pastas individualmente.

Desta forma, foi possível categorizar as imagens, documentos, vídeos e arquivos de texto, dentre outros, contidos no cartão de memória, que pudessem ser úteis a investigação.

9 CONSIDERAÇÕES FINAIS

Diversos são os métodos que um Especialista Forense pode utilizar para a analisar e extrair dados de um dispositivo móvel, mas antes de tudo, o especialista deve conhecer as características da plataforma Android antes de iniciar uma investigação.

Portanto, com os conhecimentos necessário sobre a computação forense em dispositivos móveis com o sistema operacional Android instalado, foi utilizado ferramentas capazes de auxiliar o Analista Pericial no processo de extração de informações dos dispositivos.

Esta abordagem consiste na aquisição e análise de mensagens SMS de um dispositivo móvel com os sistemas operacionais Android, listando a maneira correta para efetuar essa análise. Além, de abordar o método de extração manual das informações em um smartphone, o que envolve simplesmente percorrer os dados em um dispositivo móvel e visualiza-los, diretamente no aparelho.

No contexto Forense, o trabalho aborda assuntos como metodologia e a possibilidade de aquisição de dados em smartphones com SO Android instalado, bem como a utilização do SDK Manager e o ADB, possibilitando direcionar o foco da pesquisa para da ferramenta "*AFLogical-OSE"*, disponível no sistema operacional "*Cyborg Hawk"* utilização do sistema operacional. Também, o presente trabalho mostra como o profissional da computação forense pode fazer uso de outras ferramentas "*Open-Source",* disponível na internet para análise das mídias encontradas no dispositivo, criando uma estação de trabalho mais completa.

Por fim, foram explorados técnicas e procedimentos para aquisição e exame de dados com a finalidade de abstração de mensagens (SMS) de smartphones com sistema operacional Android a partir de um contexto comum, no qual o aparelho, no momento da apreensão, encontra-se ligado, sem restrições de acesso e sem permissões de super usuário, além da possibilidade de utilização de outras ferramentas.

BIBLIOGRAFIA

BARBOSA, P. H. **Análise das Permissões e Violações de Privacidade em Aplicações para Android**. 2017. 52 f. Trabalho de Conclusão de Curso (Graduação) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2017.

BRASI. Senado Federal, **Código de processo penal**. Brasília, DF. 2017. 187 f. Disponível em: <http://www2.senado.leg.br/bdsf/bitstream/handle/id/529749/codigo_de_processo_penal_1ed.pdf >. Acesso em 08 de agosto de 2018.

CHAINHO, F. N G. **Plataforma parametrizável para análise forense em dispositivos móveis**. 2014. 132 f. Mestrado em Engenharia de Segurança Informática (Mestrado) - Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Beja, Beja. 2014.

CLEMENTE, R. B. **Técnicas da forense computacional para coleta, identificação e preservação de evidência digital na análise lógica em ambiente Windows (NTFS)**. Cuiabá, 2009. 125 p.

DEMARTINI, F. **Brasil já tem mais de um smartphone ativo por habitante**. Brasil: Canal Tech, 20 de abril de 2018. Disponível em: <https://canaltech.com.br/produtos/brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-112294/>. Acesso em 24 de outubro de 2018.

DEVELOPERS Android Studio: **Atualizar o IDE e o SDK Tools**. 2018. Disponível em: <https://developer.android.com/studio/intro/update.html>. Acesso em 21 de abril de 2018.

ELEUTÉRIO, P. M., & MACHADO, M. P. **Desvendando a Computação Forense**. 5 Ed. São Paulo, Brasil. Novatec, 2010. 198 p.

FOLTRAN, R., & SHIBATTA, L. **A ciência forense e as principais áreas auxiliares**. In: SANTOS, F. P. DOS, VIVAN, R. H. F. (Org.). **Atenção ao idoso**: **ação multiprofissional em saúde**. Londrina: EdUnifil, 2011. 153 p. Disponível em: <http://unifil.br/portal/images/pdf/documentos/livros/atencao-ao-idoso.pdf#page=16>. Acesso em 19 de julho de 2018.

FOTOFORENSICS. **METADATA ANALYSIS**. 2012. Disponível em: <http://fotoforensics.com/tutorial-meta.php />. Acesso em 19 de julho de 2018.

GOMES, H. S. **Brasil tem 116 milhões de pessoas conectadas à internet**, **diz IBGE**. Brasil: G1, 21 de fevereiro de 2018. Disponível em: <https://g1.globo.com/economia/tecnologia/noticia/brasil-tem-116-milhoes-de-pessoas-conectadas-a-internet-diz-ibge.ghtml>. Acesso em 24 de outubro de 2018.

HOOG, A. **Android Forensics Investigation**, **Analysis, and Mobile Security for Google Android**. USA: Elsevier, 2011. 372 p.

INFORMATION SECURITY AND FORENSICS SOCIETY (ISFS). **Computer Forensics - Part 2: Best Practices,** 2009. Disponível em: <http://www.isfs.org.hk/publications/ISFS_ComputerForensics_part2_20090806.pdf>. Acesso em 23 de agosto de 2018.

JANSEN, W., AYERS, R., & BROTHERS, S. **Guidelines on Cell Phone Forensics**. Boston, MA. 2010. 687 p. Disponível em: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-101.pdf>.

JUNIOR, O. D., & CLARO, D. B. Uma análise do reconhecimento textual de nomes de pessoas e organizações na computação forense. In: **PROCEEDING OF THE SIXTH INTERNATIONAL CONFERENCE ON FORENSIC COMPUTER SCIENCE – ICOFCS 2011**, 2011, Brasilia, 210 p. Disponível em: <http://icofcs.org/2011/ICoFCS2011-PP01.pdf>. Acesso em 23 de julho de 2018.

MESQUITA, P. L. **Desafios da forense em dispositivos móveis**. Brasil, SC, 2018. Disponível em: <https://riuni.unisul.br/bitstream/handle/12345/4807/pablo_lopes_mesquita-%5b48366-11299-2-688228%5d46653-48366ad4_artigo_v1.3.pdf?sequence=1&isallowed=y>. Acesso em 19 de julho de 2018.

NELSON, B., PHILIPS, A., & STEUART, C. **Guide to Computer Forensics and Investigations**. Estados Unidos, Cengage, 2010, 663 p. Disponível em: <http://ebook.eqbal.ac.ir/Security/Forensics/Guide%20to%20Computer%20Forensics%20and%20Investigations.pdf>. Acesso em 28 de julho de 2018.

NETO, L. P. **Crimes Cibernéticos**: **Necessidade de uma legislação específica no Brasil**. João Pessoa, 2009. 39 p. Disponível em: <http://www.fespfaculdades.com.br/painel/uploads/arquivos/trabarquivo_11052010080523_linfolfo.pdf>. Acesso em 28 de julho de 2018.

SANCHES, P., TÉOFILO, A., CERQUEIRA, F., SILVA, J. A., & PAULINO, H. **Computação Distribuída em Redes Formadas por Dispositivos Móveis**, 2014.

SILVA, G. M. **Guia Foca GNU/Linux**. dezembro de 2010. Disponível em: <http://www.guiafoca.org/>. Acesso em 24 de maio de 2018.

TAMMA, R., & TINDAL, D. **Learning Android Forensics**. Birmingham, UK: Packt., 2015.

TAMMA, R., SKULKIN, O., MAHALIK, H., & BOMMISETTY, S. **Practical Mobile Forensics** (Third Edition ed.). Birmingham: Packt, 2018.

VIEIRA, J. V. **Forense digital em dispositivos móveis com sistema operacional Android**, 2013. Disponível em: <https://pt.slideshare.net/ValdemirVieiraJunior/tgitexto-final>. Acesso em 28 de agosto de 2018.