Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

ANÁLISE FORENSE DIGITAL À APLICAÇÃO MÓVEL MIFIT

JOSÉ CARLOS FERREIRA FRANCISCO

Leiria, Setembro de 2022

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

ANÁLISE FORENSE DIGITAL À APLICAÇÃO MÓVEL MIFIT

JOSÉ CARLOS FERREIRA FRANCISCO
Número: 2202274

Realizado no âmbito de Projeto sob orientação do Professor Doutor Miguel Monteiro de Sousa Frade e do Professor Doutor Patrício Rodrigues Domingues.

Leiria, Setembro de 2022

AGRADECIMENTOS

Gostaria de agradecer aos meus orientadores, Professor Doutor Miguel Monteiro de Sousa Frade e do Professor Doutor Patrício Rodrigues Domingues pela experiência e conhecimento transmitido que me encorajaram durante todo o processo de desenvolvimento deste projeto.

Deixo também uma nota de agradecimento aos meus familiares, amigos e colegas de trabalho que de alguma forma me ajudaram e acompanharam nesta etapa académica.

RESUMO

MiFit é uma aplicação para smartphone onde os utilizadores podem monitorizar parâmetros relacionados com a saúde e bem estar. Os dados são gerados recorrendo a *wearables*, onde se incluem relógios e pulseiras inteligentes. Apesar destes *wearables* possuírem um poder de processamento considerável, o espaço de armazenamento é limitado, havendo a necessidade de exportar dos dados para a aplicação no dispositivo gestor (*smartphone*). É na aplicação MiFit que são apresentados os dados numa interface gráfica simples e intuitiva, no entanto, nem todos os dados e meta-dados são mostrados ao utilizador final, podendo estes serem relevantes do ponto de vista forense.

Por se revelar uma aplicação que é usada em larga escala, no âmbito deste projeto foram estudados os artefactos forenses digitais da aplicação MiFit no dispositivo gestor. A análise estática passou passou pelo escrutínio das bases de dados e `Shared Preferences` considerados relevantes do ponto de vista forense. Este relatório descreve ainda as principais características e funcionalidades das versões previamente lançadas das pulseiras MiBand, com o intuito de perceber a evolução das mesmas. A pulseira usada foi a MiBand 6, revelando-se ser um *wearable* bastante completo, capaz de gerar informações sobre batimentos cardíacos, atividades desportivas e respetivas coordenadas Global Positioning System (GPS), monitorização do sono, stress só para enumerar alguns.

No decorrer dos trabalhos surgiu a necessidade de implementar uma plataforma, MiFitAnalyzer, que objetiva gerar artefactos forenses de forma automática baseado nos conteúdos da aplicação MiFit ou uma cópia do armazenamento do dispositivo Android. A ferramenta gera um relatório dinâmico, com toda a informação relevante encontrada. Foram ainda implementados dois módulos externos para o software *open-source* Autopsy, mundialmente usado e reconhecido na temática da análise forense.

ABSTRACT

MiFit is a smartphone application where users can monitor parameters related to health and wellness. The data is generated using wearables, including smartwatches and bracelets. Although these wearables have considerable processing power, storage space is limited, and there is a need to export the data into the application on smartphone. MiFit application have an intuitive graphical interface, however, not all data and metadata are shown to the end user. This data may be relevant from a forensic perspective.

Due the application is used on a large scale, in this project we studied the digital forensic artifacts of the MiFit application on the management device. The static analysis went through the scrutiny of the databases and Shared Preferences considered relevant from a forensic point of view. This report also describes the main features and functionalities of the previously released versions of MiBand bracelets in order to understand their evolution. The bracelet used was the MiBand 6, revealing itself to be a wearable capable of generating data about heart rate, sports activities and their GPS coordinates, sleep monitoring, stress just to name a few.

This project also includes the development of a platform, MiFitAnalyzer, which aims to generate forensic artifacts automatically based the contents of the MiFit application or a copy of the device's storage Android. The tool generates a dynamic report with all the relevant information found. Two external modules were also implemented for the open-source Autopsy, a globally used and recognized forensic analysis software.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE LISTAGENS

LISTA DE ABREVIATURAS

ADB	Android Debug Bridge.
AOT	Ahead-of-time.
API	Application Programming Interface.
APK	Android Package.
ART	Android runtime.
AWT	Abstract Window Toolkit.
BLE	Bluetooth Low Energy.
CLI	Command Line Interface.
CNAME	Canonical Name.
CSV	Comma Separated Values.
DER	Diagramas de entidade e relacionamento.
DML	Data Manipulation Language.
DND	Do Not Disturb.
DNS	Domain Name System.
FAMA	Forensic Analysis for Mobile Apps.
GPS	Global Positioning System.
HTML	HyperText Markup Language.
IEC	Comissão Eletrotécnica Internacional.
iOS	iPhone Operating System.

IP	Ingress Protection.
IPL	Instituto Politécnico de Leiria.
JSON	JavaScript Object Notation.
JVM	Java Virtual Machine.
KML	Keyhole Markup Language.
LABCIF	Laboratório de Cibersegurança e Informática Forense.
LED	Light Emitting Diode.
MD5	Message Digest Algorithm.
MobSF	Mobile Security Framework.
NFC	Near-field Communication.
OLED	Organic Light Emitting Diodes.
OSDFCon	Open Source Digital Forensics Conference.
PAI	Personal Activity Intelligence.
PKI	Public Key Infrastructures.
SDK	Software Development Kit.
SMS	Short Message Service.
SO	Sistema Operativo.
SQL	Structured Query Language.
SSL	Secure Sockets Layer.
TTL	Time to live.

USB Universal Serial Bus.

VM Máquina Virtual.

XML Extensible Markup Language.

1

INTRODUÇÃO

A crescente preocupação por um estilo de vida ativo e saudável leva com que a população procure soluções para monitorizar o seu estilo de vida e posteriormente definir metas e objetivos de o tornar mais saudável. Essa monitorização pode ser feita recorrendo a *Fitness Trackers* apresentando-se principalmente sob forma de pulseiras e relógios inteligentes, calçado ou acessórios corporais (Henriksen et al., 2018).

Inicialmente estes vestíveis [1] apresentavam-se simples, com funcionalidades básicas e uma interação com o utilizador limitada. Atualmente são dispositivos mais avançados recorrendo maioritariamente a sensores aumentando as suas funcionalidades e capacitando-os de operar de forma autónoma durante vários dias.

Complementando este tipo de dispositivos, tipicamente está associado um dispositivo gestor, surgindo vulgarmente sob forma de aplicação para dispositivos móveis (*smartphones)*, controlando-o e tirando partido as funcionalidades presentes neste.

A MiFit apresenta-se como uma aplicação direcionada à saúde e bem-estar. Desenvolvida pela empresa chinesa Xiaomi, a aplicação foi lançada em 2014 e visa recolher dados de saúde e *fitness* resultantes da conexão com outros dispositivos. Disponível para *smartphones*, à data deste documento a MiFit ocupa a quarta posição das aplicações de saúde e *fitness* mais descarregada em todo o mundo (Ceci, 2022a). Em Janeiro de 2022, as principais aplicações móveis de *fitness* registaram quase 17 milhões de downloads em todo o mundo (Ceci, 2022b).

A empresa Xiaomi é uma das maiores produtoras de *wearables* em todo o mundo, ocupando consecutivamente a segunda posição no maior número total de exportações de dispositivos deste tipo em todo o mundo de 2014 a 2021 (Laricchia, 2022). Da vasta gama de dispositivos fabricados pela empresa chinesa, neste projeto foi usada a pulseira inteligente MiBand. Trata-se de um dispositivo com um número apreciável de sensores e funcionalidades, com elevada popularidade, em parte devido ao seu custo a rondar os 40 euros e às funcionalidades que disponibiliza.

1 Entende-se como um *vestível* qualquer tipo de dispositivo eletrónico projetado para ser usado no corpo do utilizador

A popularidade e o facto dos dispositivos vestíveis estarem sempre junto dos respetivos utentes, levam a que possam ser uma preciosa fonte de informação no contexto de um cenário de investigação. A principal motivação deste trabalho é precisamente a identificação e o estudo dos artefactos forenses criados pelo uso de um par MiBand/ MiFit.

Convergindo os dados gerados pelas pulseiras MiBand e da MiFit, destacam-se as funcionalidades de:

- Registo do número de passos diários
- Monitorizar o sono
- Registo do ritmo cardíaco
- Medição/monitorização do nível de saturação de oxigénio no sangue
- Identificar momentos de stress
- Armazenar parâmetros associados à prática de atividade desportiva
- Receber notificações provenientes do dispositivo gestor
- Realizar transações por Near-field Communication (NFC)

Durante o decorrer do desenvolvimento deste projeto, a aplicação denominada de MiFit viu o seu nome alterado para `Zepp Life`. Apesar da alteração, os dispositivos anteriormente lançados permanecem com as funcionalidades disponíveis, como é o caso da pulseira MiBand 6 usada neste projeto. Apesar de se ter estudado também a aplicação Zeep Life, neste documento optou-se por usar a nomenclatura MiFit, cujo a estrutura se revelou semelhante à aplicação estudada.

1.1 OBJETIVOS E MOTIVAÇÃO

A análise desenvolvida no âmbito deste trabalho passa pela identificação de artefactos forenses, analisando os conteúdos presentes no dispositivo inerentes à aplicação MiFit. Outro objetivo deste trabalho é o desenvolvimento de software para deteção e interpretação dos artefactos forenses do binómio MiBand/MiFit.

Em suma, os principais objetivos deste projeto são:

- Análise aos conteúdos da aplicação MiFit quando acopolada a uma MiBand 6
- Extração de informação relevante do ponto de vista forense da aplicação MiFit para sistemas Android

- Entender interações entre os *wearables* e os dispositivos gestores (*smartphones*)
- Implementação de uma ferramenta que torne mais célere o processo de análise dos conteúdos extraídos
- Integração da ferramenta acima mencionada com o software de análise forense Autopsy

1.2 CONTRIBUTOS

Como principais contributos deste projeto destaca-se:

- Documentação da análise forense à aplicação MiFit para sistemas Android
- Transmissão de conhecimento relativo às pulseiras MiBand, amplamente usadas como *Fitness Tracker* pelos seus utilizadores
- Disponibilização de uma ferramenta *open-source* capaz de gerar artefactos forenses que torne fácil a análise dos conteúdos da aplicação MiFit[2]
- Módulo de análise forense e construção de relatório dinâmico para o software Autopsy[3]

1.3 ESTRUTURA DO DOCUMENTO

Este documento está estruturado em oito capítulos. O presente capítulo introduziu o documento, definindo o propósito do projeto e os contributos relevantes para a comunidade forense.

O Capítulo 2 aborda o trabalho relacionado com a temática deste prejeto.

O Capítulo 3 explana os conceitos e tecnologias consideradas relevantes para o desenvolvimento deste projeto.

O Capítulo 4 aborda as pulseiras inteligentes MiBand. Tendo sido a MiBand um *wearable* essencial no desenrolar deste projeto, o capítulo enumera a evolução deste dispositivo e as suas principais funcionalidades.

O Capitulo 5 aborda a investigação à aplicação MiFit para o sistema Android, a metodologia utilizada e os resultados relevantes obtidos.

2 https://github.com/98jfran/MifitAnalyzer
3 https://github.com/98jfran/MifitAnalyzer

O Capitulo 6 aborda com detalhe a ferramenta de análise forense Autopsy, enquadrando-a no âmbito do projeto desenvolvido. Este Capitulo ainda explana a extensibilidade disponibilizada pelo software, fortemente usada pela comunidade forense.

O Capítulo 7 documenta a plataforma MiFitAnalyzer que objetiva a análise dos conteúdos gerados pela aplicação MiFit. Este capitulo explica o funcionamento da plataforma, incluindo a sua arquitetura geral e a integração que esta oferece com o software Autopsy.

Por fim, o Capitulo 8 fornece um breve resumo do trabalho apresentado, delineando possíveis passos a seguir para um trabalho futuro.

2

TRABALHO RELACIONADO

Este capítulo apresenta o trabalho relacionado da análise forense a aplicações moveis e a dispositivos vestíveis.

Para o efeito, este capítulo analisa estudos de aquisição forense para dispositivos móveis e estudos que focam a análise digital forense de dispositivos MiBand. O capítulo apresenta ainda estudos sobre a comunicação dos dispositivos vestíveis e *smartphones*, e formas de extração de dados baseadas em Application Programming Interface (API) e serviços *cloud*.

2.1 AQUISIÇÃO FORENSE

A interação com a tecnologia é constante, deixando uma pegada digital muitas vezes permanente e reveladora das atividades normais dos utilizadores com os dispositivos. Quando confrontados com um incidente digital, os peritos forenses baseiam parte do seu trabalho na aquisição de dados através do dispositivo.

Neste sentido, Sathe e Dongre (2018) explanam técnicas forenses para a aquisição de dados em dispositivos móveis. Tradicionalmente, os dados podem ser adquiridos com recurso a técnicas de aquisição por software ou hardware, categorizando-se também em tipos de aquisição lógica ou física.

É o perito forense o responsável por tomar a decisão mais adequada quanto ao método de aquisição forense a usar perante o incidente. Os autores apresentam comparações entre os diferentes procedimentos e softwares usados, concluindo que as técnicas físicas baseadas em software funcionam melhor porque não causam danos permanentes ao dispositivo, no entanto, realçando que nenhuma das estratégias aplicadas consegue fornecer todo o conhecimento do dispositivo. Por conseguinte, em muitos casos é necessário utilizar mais do que um método para alcançar os resultados desejados.

2.2 ESTUDOS ÀS PULSEIRAS MIBAND

Como descrito na introdução deste documento, o constante crescimento da preocupação por um estilo de vida saudável, levam a adoção de medidas que procurem beneficiar o bem estar físico e mental, tendo como exemplo e prática de exercício físico, monitorização do sono, adoção de hábitos alimentares mais saudáveis, só para enumerar alguns.

A constante monitorização dos parâmetros de saúde tornou-se uma necessidade. Os *wearables*, ainda que passíveis de uma taxa de erro nas suas medições (Paradiso et al., 2020), são capazes de medir de forma fiável o ritmo cardíaco, número de passos, distância e duração do sono, que podem ser utilizados como indicadores eficazes de avaliação da saúde. (Xie et al., 2018). Vários estudos foram realizados no *wearable* supracitado neste documento, a pulseira inteligente MiBand, conforme descrito de seguida.

Nesse sentido, e por se relacionar com a temática deste projeto, Kang et al., 2020 fizeram um levantamento das funcionalidades principais da MiBand 2, levando a cabo uma análise forense aos conteúdos da aplicação MiFit para essa versão da pulseira. Os autores exploraram as principais as bases de dados SQLite3 que suportam a aplicação, identificando tabelas e colunas que suportam possíveis artefactos forenses a serem extraídos baseados nos dados armazenados. A maioria dos dados gerados pela MiBand e persistidos pela aplicação MiFit são feitos de forma automática de acordo com os movimentos do utilizador, no entanto, alguns desses dados podem ser apagados ou alterados de forma manual, seja para os completar ou para os corrigir por um erro de leitura, por exemplo. Os autores exemplificam com dados capturados durante período do sono e da atividade do utilizador.

Os dados da atividade não podem ser modificados, mas podem ser apagados, deixando um rasto de que estes foram removidos. Os autores enaltecem que cabe ao investigador forense encontrar o contexto em que o utilizador modificou ou apagou estes registos.

2.3 COMUNICAÇÃO DOS WEARABLES

À semelhança do trabalho realçado anteriormente, Hantke e Dewald, 2020 abordaram a pulseira da Xiaomi MiBand 2, onde apresentam como podem ser obtidos e analisados dados provenientes de pulseiras de *fitness* seguindo uma abordagem forense.

Estes *wearables* tipicamente fornecem apenas funções básicas quando comparados com o dispositivo gestor por se tratarem de dispositivos de pequena capacidade de memória e processamento. Estes dispositivos comunicam com frequência com o dispositivo gestor usando protocolos de consumo energético baixo e largura de banda reduzida, como é o caso do Bluetooth Low Energy (BLE).

Os autores começaram por estudar as comunicações entre o dispositivo gestor - smartphone Android Samsung Galaxy S4 - e a pulseira inteligente MiBand 2 usando o protocolo BLE. O estudo das comunicações entre o smartphone e pulseira estudada não é novo. Foi baseado no artigo publicado por Nikishaev, 2018 que estes autores basearam toda a sua pesquisa relacionada com esta temática. Antes de intercetar qualquer transmissão de dados entre os dispositivos, é necessário passar por um processo de autenticação que consiste na troca de pacotes onde constam os pacotes encriptados com a chave acordada. Interpretar os dados desconhecendo a estrutura que os suporta foi uma das dificuldades encontradas durante todo o processo.

A Listagem 1 mostra um exemplo dos dados dados obtidos.

Listagem 1: Dados interceptados entre MiBand e o smartphone

```
1     Raw heart: 02102d8c348c448c458c3d8c428c488c 16
2     Raw heart: 0218468c418c3d8c468c3f8c398c418c 16
3
4     Realtime heart: 93
5
6     Raw heart: 0220408c448c3f8c428c498c3c8c3d8c 16
7     Raw heart: 02283d8c398c488c3e8c468c488c328c 16
8
9     Realtime heart: 99
10
11    Raw heart: 0230438c408c378c3a8c318c458c388c 16
12
13    Realtime heart: 102
14
15    Raw heart: 02404f8c408c458c428c4d8c558c4d8c 16
16    Raw heart: 02483e8c3b8c3f8c348c398c318c428c 16
17
18    Realtime heart: 98
19
20    Raw heart: 02504c8c428c5e8c4f8c588c498c558c 16
21    Raw heart: 0258478c458c3c8c4e8c3f8c468c4d8c 16
22
23    Realtime heart: 100
24
25    Raw heart: 0260518c4d8c4f8c4b8c4f8c528c458c 16
26    Raw heart: 0268408c3f8c538c4d8c408c548c598c 16
27
```

```
28      Realtime heart: 102
29
30      Raw heart: 0278418c508c4e8c548c588c468c498c 16
31      Raw heart: 0280368c328c2e8c3c8c338c308c3f8c 16
32
33      Realtime heart: 101
```

Se convertermos a *string* com um tamanho 2 bytes, obtemos 7 medidas numéricas correspondentes às leituras do sensor cardíaco, como mostram as linhas indicadas com o prefixo `Realtime heart`. Os autores desenvolveram trabalho no âmbito da análise do dispositivo gestor, identificaram a hierarquia de ficheiros e organização de diretorias da aplicação MiFit para a versão estudada. Para terminar, Hantke e Dewald (2020) procuraram perceber as chamadas aos serviços providenciados pela aplicação usando APIs. Analisando o tráfego de rede usando a aplicação Burp Suite[1], os autores constataram que chamadas aos servidos de Google APIs, Facebook e Huami API, tendo sido estas ultimas chamadas negadas de obter acesso à API.

2.4 EXTRAÇÃO DE DADOS POR SERVIÇOS E API

Independentemente do objetivo principal deste projeto estar orientado à análise da aplicação MiFit recorrendo somente ao existente no dispositivo gestor, uma publicação na página DEV[2] despertou interesse.

A publicação documenta vários procedimentos não oficiais para a extração dos conteúdos associados aos serviços da MiFit recorrendo aos serviços *cloud* (Oh, 2021).

Para que os dados sejam exportados para uma folha de calculo (Google Sheets[3]), é necessária preparação prévia para incluir os serviços de autenticação. De realçar deve ser adicionada a dependência do método de autenticação `OAuth2`[4] amplamente usado por aplicações web e mobile, como documentado no artigo.

Configurados os serviços de autenticação, o autor disponibiliza um *script* Python usado para aceder à API da MiFit e obter os dados.

Com o método de extração de dados recorrendo aos serviços da MiFit, o autor conseguiu recolher os seguintes dados:

- Passos

1 https://portswigger.net/burp
2 https://dev.to
3 https://www.google.com/sheets
4 https://github.com/googleworkspace/apps-script-oauth2

- Distância de caminhada e corrida
- Ritmo cardíaco
- Duração do sono
- Pontuação associada ao desporto
- Stress

Esta abordagem de extração de dados tornar-se-á eficaz caso o utilizador use o método de autenticação da Google e que sejam conhecidas as credenciais de acesso ou o *token* de autenticação na conta MiFit.

2.5 SUMÁRIO

Esta capitulo sintetizou alguns dos trabalhos e estudos relacionados com a temática deste projeto. Foram abordadas temáticas como métodos de aquisição forense, estudos sobre protocolos de comunicação Bluetooth, serviços e API. O próximo capítulo aborda conceitos e tecnologias empregues no âmbito deste projeto.

3

CONCEITOS E TECNOLOGIAS

O projeto apresentado envolveu a aplicação de um conjunto de métodos e tecnologias com o intuito de resolver os problemas apresentados. Este capítulo apresenta algumas dessas tecnologias e os métodos aplicados associados à informática forense.

3.1 ANDROID DEBUG BRIDGE

A ferramenta ADB[1] incluída no Android Software Development Kit (SDK) é uma opção segura e viável para estabelecer comunicação entre um computador e um dispositivo Android. Esta comunicação pode ser estabelecida via Universal Serial Bus (USB) ou Wi-Fi.

O ADB é tipicamente usado na forma de Command Line Interface (CLI), disponibilizando um vasto conjunto de funcionalidades. A Tabela 1 lista os principais comandos usados no âmbito deste projeto.

COMANDO	DESCRIÇÃO
adb devices -l	listar dispositivos conectados
adb install *pt.package*	instalar *packages* no dispositivo
adb pull *remote/path local/path*	enviar ficheiros para o dispositivo
adb push *local/path remote/path*	extrair ficheiros do dispositivo
adb shell *comando*	executar comando no dispositivo

Tabela 1: Principais comandos ADB

3.2 JADX

Uma das linguagens principais para o desenvolvimento de aplicações para o Sistema Operativo (SO) Android é o Java. Sendo o Java uma linguagem que não compila nativamente para código interpretável por um processador, é necessária

1 https://developer.android.com/studio/command-line/adb

uma Máquina Virtual (VM) que transforma para formato intermediário, chamado de `bytecode`. A VM nos sistemas Android, denominada de `Dalvik`, baseia-se no *kernel* Linux, usando as suas potencialidades como *threading* e gestão de memória (Ehringer, 2010). A tecnologia `Dalvik` tem sido substituída, dando lugar ao Android runtime (ART), mais otimizada apresentando a compilação Ahead-of-time (AOT) que melhora o desempenho da aplicação (Android Documentation, 2020).

Durante o processo de compilação, algumas informações escritas originalmente pelo programador, onde se incluem nomes de variáveis e funções por exemplo, são perdidas ou ofuscadas. Do processo de compilação do código fonte da aplicação para o *bundle* final, resulta um ficheiro único em formato Android Package (APK).

A aplicação JADX[2] atua como um descompilador e desofuscador, oferecendo um ambiente gráfico para explorar o código fonte de um APK ou de um ficheiro `.dex` gerado no processo de compilação.

A Figura 1 mostra a aplicação JADX durante um processo de análise aos ficheiros binários da aplicação MiFit.

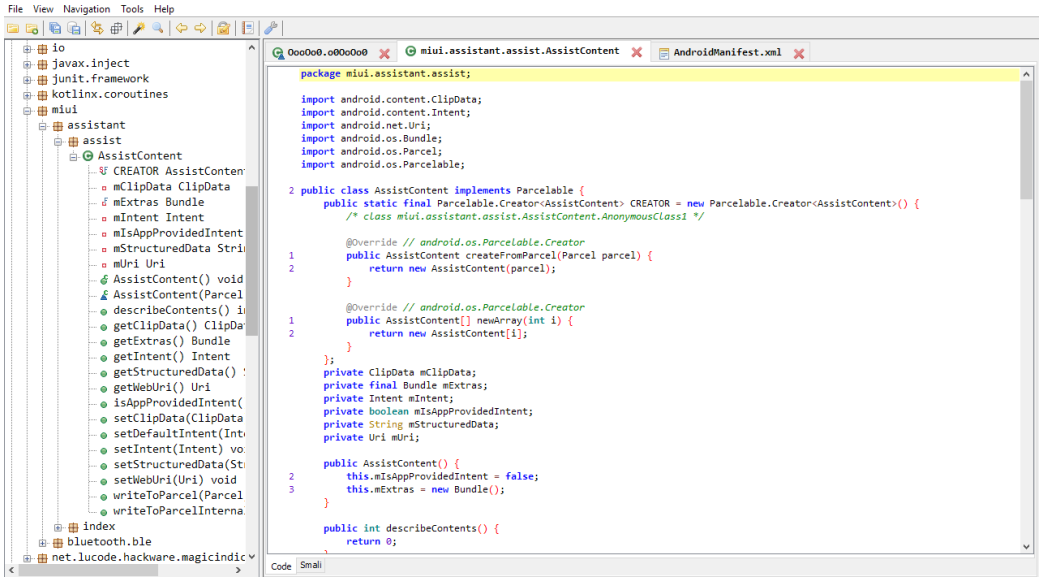

Figura 1: Ferramenta JADX

2 https://github.com/skylot/jadx

3.3 HXD

Todos os ficheiros são passiveis de ser representados de forma binária. O facto de estes ficheiros possuírem estruturas proprietárias ou desconhecidas, fazem dos editores de texto uma ferramenta pouco útil nestas circunstâncias.

A ferramenta HxD[3] é um visualizador e editor de ficheiros em formato hexadecimal. No decorrer deste projeto, esta ferramenta foi utilizada para visualizar alguns dos ficheiros binários da aplicação MiFit.

A Figura 2 mostra a aplicação HxD durante um processo de análise aos ficheiros binários da aplicação MiFit.

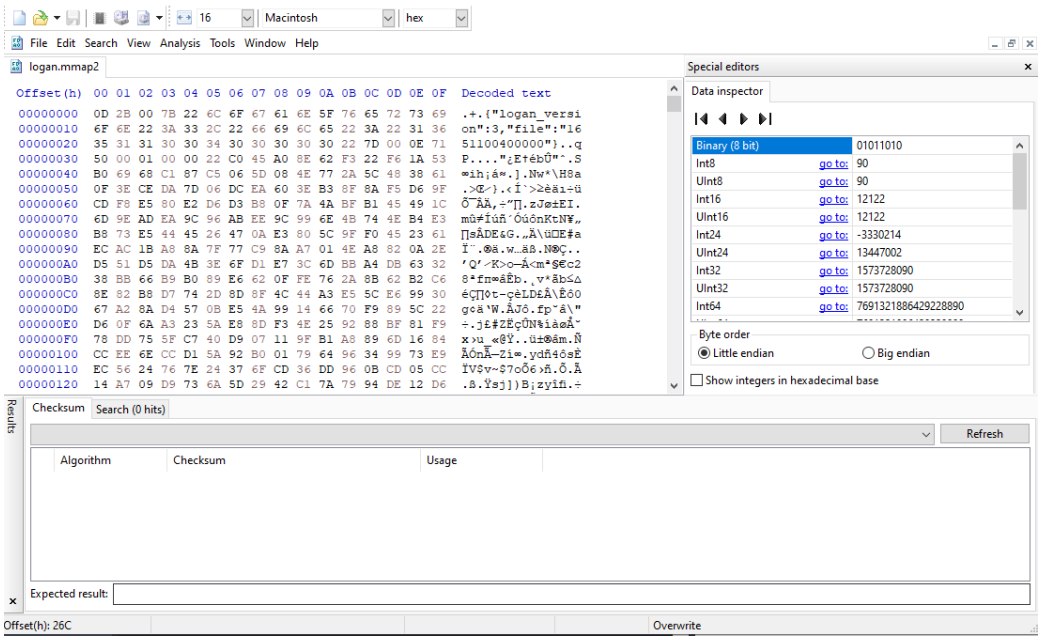

Figura 2: Ferramenta HxD

3.4 DB BROWSER FOR SQLITE

O SQLite3[4] é motor de bases de dados utilizado no desenvolvimento de aplicações moveis. Tipicamente a base de dados é representado num único ficheiro, podendo esta recorrer a ficheiros temporários para o seu correto funcionamento. No processo de visualização e análise dos conteúdos presentes nas bases de dados do tipo SQLite3, foi usado o software `DB Browser for SQLite`[5].

3 https://mh-nexus.de/en/hxd/
4 https://www.sqlite.org/
5 https://sqlitebrowser.org/

3.5 SCHEMACRAWLER

Schemacrawler[6] é uma ferramenta que permite descobrir a arquitetura de uma base de dados. A ferramenta permite ainda gerar multiplos Diagramas de entidade e relacionamento (DER), usados para o estudo das relações entre as bases de dados que integram a aplicação estudada.

3.6 SUMÁRIO

Este capitulo abordou algumas ferramentas e tecnologias cruciais na resolução de problemas durante o desenvolvimento do projeto. Foi destacada a interface de depuração ADB como o principal meio de extração dos conteúdos da aplicação de um dispositivo móvel durante o decorrer dos testes. Foram ainda descritas algumas ferramentas de análise empregues no contexto deste trabalho.

6 https://www.schemacrawler.com/

4

MIBAND - SMARTBAND

Este Capítulo descreve a pulseira dita inteligente MiBand dado a relevância que ocupa neste projeto. O capítulo contextualiza a pulseira, e foca as principais funcionalidades e características, bem como funcionamento do dispositivo vestível.

4.1 O FENÓMENO E EVOLUÇÃO

MiBand é uma pulseira inteligente (*wearable*) produzida pela marca a chinesa Xiaomi[1]. Rapidamente ganhou fama pelo seu tamanho reduzido e as suas numerosas funcionalidades. A pulseira tem sido referência para outros fabricantes, aumentando assim a diversidade de dispositivos no mercado e os potenciais utilizadores deste tipo de *wearables*.

Idealmente, o utilizador usa diariamente a pulseira, permitindo que esta recolha informações relacionadas com a atividade desportiva, saúde e bem-estar. A ligação entre a pulseira e o dispositivo gestor (*smartphone*) é assegurada pela tecnologia BLE, caracterizada como uma tecnologia fiável de curto alcance e de pouco consumo energético, ideal para dispositivos vestíveis de tamanho reduzido como é o caso da MiBand.

A primeira versão da MiBand data de 2014, sendo que à data de escrita deste capítulo encontra-se na sexta edição, existindo fortes rumores do lançamento ainda em 2022 da sétima edição. Para melhor entender a evolução da MiBand, focaremos os pontos principais de cada versão como mostram as secções seguintes.

1 https://www.mi.com/

4.1.1 *MiBand 1 (2014)*

Figura 3: Pulseira inteligente MiBand 1

Fonte: https://www.mi.com/global/miband

Apesar da sua aparência básica quando comparada com as versões mais atuais, na sua a primeira edição, MiBand contava com a funcionalidade de monitorizar o sono e atividades desportivas como principais funcionalidades. Os dados recolhidos apenas são visíveis na aplicação móvel (Android e iPhone Operating System (iOS)) no dispositivo gestor.

Quanto ao seu aspeto exterior, esta primeira versão não possui qualquer *display*, apresentando apenas três indicadores Light Emitting Diode (LED), representativos do nível de bateria restante da pulseira.

4.1.2 *MiBand 1s (2015)*

Em 2015, a Xiaomi apresenta a MiBand 1s, exteriormente semelhante à primeira versão (MiBand 1) mas com melhorias significativas nas funcionalidades. Mantendo todas as funções da sua antecessora, na versão MiBand 1s foi introduzido o sensor óptico de medição da frequência cardíaca. A adição deste sensor permite monitorizar parâmetros associados ao batimento cardíaco em tempo real efetuando leituras regulares do batimento cardíaco, enriquecendo os dados recolhidos nas atividades diárias e desportivas.

4.1.3 *MiBand 2 (2016)*

Figura 4: Pulseira inteligente MiBand 2

Fonte: https://www.mi.com/global/miband2/

Na segunda iteração da pulseira, MiBand 2 reuniu as melhores características das suas antecedentes num hardware e *design* completamente renovado.

A grande novidade na MiBand 2 foi a sua tela com tecnologia Organic Light Emitting Diodes (OLED), algo novo para as pulseiras MiBand. A presença da tela permitiu melhor a interação do utilizador com o *wearable*, estendendo as suas funcionalidades. Nesta versão já é possível ver notificações em tempo real, a contagem dos passos diários do utilizador e frequência cardíaca. O botão localizado na parte inferior permite alternar entre os menus.

A resistência da pulseira também foi melhorada, tendo sido adicionada proteção fisica, nomeadamente suporte para a certificação IP67[2] da escala Ingress Protection (IP) definida pela Comissão Eletrotécnica Internacional (IEC) 60529. De acordo com a norma IP67, a MiBand deve resistir a poeira e suportar imersão em água de até um metro por um período de 30 minutos.

2 https://www.iec.ch/ip-ratings

4.1.4 *MiBand 3 (2018)*

Figura 5: Pulseira inteligente MiBand 3

Fonte: https://www.mi.com/global/mi-band-3

Em 2018, a MiBand 3 apresentou-se com uma tela maior e com o botão que suportava gestos. O dispositivo continuou a expandir o leque de funcionalidades, anunciando a possibilidade de de aceder à previsão meteorológica e implementando melhorias nas funcionalidades relacionadas com a monitorização de atividades desportivas.

A resistência da pulseira foi novamente melhorada, subindo uma escala na categoria de líquidos, tendo sido atribuído um grau geral de IP68, significando que a pulseira é capaz de tolerar imersão na água de forma contínua até três metros de profundidade.

4.1.5 *MiBand 4 (2019)*

Figura 6: Pulseira inteligente MiBand 4

Fonte: https://www.mi.com/global/mi-smart-band-4

Apesar de o botão localizado na parte inferior ainda estar presente, a MiBand 4 apresenta um ecrã tátil que permite navegar e entrar nos menus da pulseira. A possibilidade de ativar o GPS quando o telefone está conectado foi introduzida nesta versão 4. Foram adicionadas algumas funcionalidades que permitem interagir com dispositivo gestor, onde se incluem o controlo de música – volume e música anterior/próxima, bem como a possibilidade de atuar como botão de disparo da câmara fotográfica do *smartphone*.

4.1.6 *MiBand 5 (2020)*

Figura 7: Pulseira inteligente MiBand 5

Fonte: https://www.mi.com/global/mi-smart-band-5

As melhorias entre a MiBand 4 e a MiBand 5 não se revelaram muito significativas quando comparadas com as iterações anteriores.

Na MiBand 5, apresentada em 2020, destaca-se a tela maior e com mais fluidez. Foi ainda adicionado o suporte para mais atividades desportivas, podendo assim gerar mais artefactos com base nesta funcionalidade.

4.1.7 *MiBand 6 (2021)*

Figura 8: Pulseira inteligente MiBand 6

Fonte: https://www.mi.com/global/mi-smart-band-6

A março de 2021 é lançada a sexta versão da MiBand. Semelhante ao que se tem verificado nas gerações anteriores, a MiBand 6 apresenta uma tela maior e com maior fluidez face à geração anterior. O botão frontal de navegação foi removido, fazendo com que a navegação entre menus seja feita unicamente recorrendo à tela tátil da pulseira.

A incorporação de um leitor ótico para medição dos níveis de oxigénio no sangue foi outra novidade nesta geração. As medições são maioritariamente feitas de forma manual, isto é, a pedido do utilizador.

Mais tarde, em setembro de 2021 é lançada uma nova versão da MiBand 6 que permite comunicações usando protocolos NFC. A integração desta tecnologia abre um novo leque de possibilidades, como automatizações de sistemas que usem esta tecnologia e a possibilidade de realizar pagamentos com a MiBand.

Em suma, a Tabela 2 lista as principais características, onde se incluem dimensões, sensores e protocolos de comunicação das três gerações mais recentes à data da escrita deste documento.

	MIBAND 6	MIBAND 5	MIBAND 4
Dimensões	47.4x18.6x12.7mm	46.95×18.15×12.45mm	46.8x17.8x12.6mm
Resolução ecrã	360x152 pixeis	126x294 pixels	120x240 pixeis
Dimensão do ecrã	1.5 polegadas	1.1 polegadas	0.95 polegadas
Sensores	Sensor de 6 eixos: Acelerómetro de 3 eixos e giroscópio de 3 eixos; Sensor do ritmo cardíaco PPG	Sensor de 6 eixos: Acelerómetro de 3 eixos e giroscópio de 3; Sensor do ritmo cardíaco PPG; Microfone MEMS digital	Acelerómetro de 3 eixos + giroscópio de 3 eixos; sensor do ritmo cardíaco PPG; Sensor de proximidade capacitivo
Conectividade	BT5.0 BLE	BT5.0 BLE	BT5.0 BLE

Tabela 2: Tabela comparativa das características da MiBand nas versões 4, 5 e 6

4.2 FUNCIONALIDADES

Esta secção relata as principais funcionalidades da pulseira inteligente MiBand.

A funcionalidades abaixo identificadas foram listadas usando as configurações originais de fábrica, não alterando nenhum componente de hardware ou software. De notar que as descrições apresentadas abaixo baseiam-se na MiBand 6, sendo que algumas delas não se enquadram nas versões anteriores.

A lista seguinte apresenta as funcionalidades abordadas nos parágrafos seguintes.

- Batimento cardíaco
- Oxímetro
- Sono
- Alarme
- Desporto
- Respiração
- Stress
- Meteorologia
- Lanterna
- Temporizador
- Pomodoro
- Cronômetro
- Notificações
- Passos
- Histórico de atividades
- Música
- PAI
- Relógio mundial
- Eventos
- Câmara
- Encontrar dispositivo
- Silêncio
- Ciclo menstrual

Batimento cardíaco. A MiBand mede os batimentos cardíacos durante a atividade diária do utilizador em determinados intervalos de tempo que podem ser configurados na aplicação MiFit. Fora deste período, é possível realizar uma medição de pulso manual.

Oxímetro. Uma das novidades que se destaca na MiBand 6 face às versões anteriores da pulseira inteligente é o sensor de nível de oxigénio no sangue. Esta funcionalidade, também conhecida por SpO2, mede a saturação de oxigénio da hemoglobina no sangue comparando a absorção da luz de diferentes comprimentos de onda através de uma parte translúcida do corpo (World Health Organization, 2021).

Sono. A MiBand analisa os parâmetros durante o sono, monitorizando e analisando-os com objetivo de ajudar e ajustar a sua qualidade. A pulseira parametriza o sono em 4 fases distintas, `Sono profundo` (*Deep sleep*), `Sono leve` (*Light sleep*), `REM` (*Rapid Eye Movement*), Tempo acordado (*Time awake*). A pulseira identifica a fase e a duração em que o utilizador se encontra durante sono. Baseado nestes parâmetros, a aplicação MiFit apresenta um resumo da análise que efetuou à qualidade do sono.

Alarme. A função de alarme funciona como um despertador. Na pulseira, é possível criar novos horários, ativar e/ou desativar um alarme que já tenha sido definido anteriormente e configurar repetições. De notar que esta funcionalidade apenas é conseguida com recurso a vibração, sendo que a pulseira não emite qualquer som.

Desporto. Um dos propósitos para a qual a MiBand foi concebida é a possibilidade de gravar parâmetros de atividades desportivas. Esta funcionalidade está disponível no separador de atividades.

O utilizador dispõe de vários perfis de desporto à escolha. Padrões como corrida, natação, ciclismo, dança, yoga, só para enumerar alguns. De realçar que o número de padrões têm sofrido alterações notáveis, podendo constatar uma preocupação por parte da Xiaomi em incluir suporte para mais tipos de atividades. Por exemplo, na versão 5, a MiBand apenas suportava 11 tipos de actividades distintas, contrastando com as 30 disponibilizadas na versão 6 (Sawh, 2021).

Á data da escrita deste documento, a pulseira MiBand 6 identifica automaticamente seis atividades – corrida exterior, passadeira, máquina de remo, elíptica, caminhada–, isto é, o utilizador não necessita de informar a pulseira que está a praticar estas atividades.

Respiração. A funcionalidade de respiração permite ao utilizador fazer exercícios respiratórios entre 1 e 5 minutos, com o intuito de relaxamento. A pulseira emite pequenos pulsos vibratórios, sendo que nos intervalos dos mesmos é pedido ao utilizador que inspire e expire de forma continua e sincronizada. No final do tempo, a pulseira indica o batimento cardíaco antes e após o exercício. De notar que esta funcionalidade não contabiliza nem regista a frequência respiratória do utilizador.

Stress. A funcionalidade de stress calcula um valor inteiro entre 0 e 100 representativo do nível de stress baseado na variação do ritmo cardíaco. Após a leitura, a pulseira MiBand apresenta o resultado obtido enviando-o para a aplicação MiFit na próxima sincronização de dados.

Meteorologia. A pulseira dá informação sobre a meteorologia no local selecionado. Este separador apresenta uma previsão para o dia inteiro e para os quatro dias seguintes.

Lanterna. Ao ativar a funcionalidade de lanterna, a pulseira coloca o brilho do ecrã com a intensidade máxima e cor branca.

Temporizador. Funcionalidade de temporizador, onde o utilizador define que um valor temporal e este é descontado ao longo do tempo quando iniciado. De notar que esta funcionalidade não armazena quaisquer dados na MiBand.

Pomodoro. Esta funcionalidade permite aplicar a técnica `pomodoro`. Esta técnica consiste na aplicação de um método de gestão do tempo baseado em intervalos definidos na pulseira (entre 5 minutos a 60 minutos) de trabalho quebrados por pausas de 5 minutos e pausas de 15 a 30 minutos após a conclusão de quatro períodos de trabalho (Collins, 2020).

Cronometro. Funcionalidade de cronometro.

Notificações. Lista de notificações emitidas pelo dispositivo gestor (*smartphone*). Esta funcionalidade requer que o utilizador forneça permissão de leitura de todas as notificações criadas/recebidas pelo smartphone. As aplicações que fazem despoletar são totalmente parametrizáveis, sendo que é o utilizador que decide de quais as aplicações quer receber as notificações. Com esta funcionalidade ativa, o utilizador recebe notificações de Short Message Service (SMS) e de aplicações configuradas (e.g., WhatsApp, Signal, ...) sempre que o smartphone estiver com a tela apagada.

Passos. Neste menu são mostradas informações relacionadas com a atividade diária do utilizador, onde se inclue: o número de passos dados no corrente dia associada à distancia e ao gasto calórico. Neste menu é ainda apresentada a contagem do número de passos dos últimos 7 dias.

Histórico de atividades. Neste separador, o utilizador pode consultar o sumário das atividades desportivas realizadas. Para cada atividade, na MiBand constam apenas informações básicas, como a data e hora, distancia percorrida e a duração.

Música. A MiBand permite controlar a música que está a tocar no dispositivo.

O utilizador tem o controlo do volume da música, pausar a reprodução e alternar para a faixa anterior e para a faixa seguinte. A MiBand não está equipada com altifalante, pelo que não é possível emitir qualquer som ou música pela pulseira.

PAI. Personal Activity Intelligence (PAI) é um índice que de atividade física e bem-estar. Baseado nas características do utilizador, como a idade, género, qualidade de descanso, batimentos cardíacos e atividade física, é atribuída uma pontuação numa escala de 0 a 125. Segundo Zisko et al., 2017 uma pontuação semanal superior a 100 reduz significativamente o aparecimento de doenças cardíacas.

A MiBand faz o cálculo do PAI, sendo possível ver a progressão ao longo do tempo.

Relógio mundial. Relógios de várias cidades mundiais. As cidades a serem mostradas podem ser adicionadas e removidas na aplicação.

Eventos. A função de eventos é uma forma fácil e rápida de registar lembretes. Estes eventos apenas podem ser adicionados e alterados recorrendo a aplicação móvel. Quando chega a hora programada, a MiBand emite um sinal vibratório, alertando o utilizador da nota adicionada previamente na aplicação.

Câmara. Controlo remoto da câmara do smartphone usando a pulseira.

Encontrar dispositivo. Esta funcionalidade permite localizar facilmente o *smartphone* com a MiBand. Para isso, a MiBand deve estar conectada ao *smartphone* por bluetooth e, após a instrução do utilizador, o *smartphone* emite um sinal sonoro que se repete de forma similar ao da receção de uma chamada. De realçar que o dispositivo tocará mesmo que este esteja no modo silencioso ou Do Not Disturb (DND).

Silêncio. Esta funcionalidade permite ao utilizador colocar o dispositivo gestor no modo silencioso. Para que está funcionalidade seja aplicada, é necessário que o modo DND esteja ativo na aplicação.

Ciclo menstrual. A funcionalidade intitulada de `cycles` ajuda a acompanhar ciclos menstruais femininos, apresentando um calendário estimado para a menstruação e períodos de ovulação. É também possível marcar um lembrete para um próximo período, e nesse dia, a MiBand encarregar-se-á de lembrar o utilizador. Por omissão, esta funcionalidade não está disponível na pulseira. É necessário ativar a funcionalidades na aplicação MiFit.

4.3 SUMÁRIO

Neste capítulo foi abordado a série de pulseiras MiBand, descrevendo as suas principais funcionalidades e o poder que estas poderão ter na recolha de artefactos forenses. Foram ainda descritos os pontos fortes de cada geração da pulseira inteligente.

O próximo capítulo foca a aplicação móvel MiFit.

5

MIFIT - APLICAÇÃO MÓVEL

MiFit é uma aplicação direcionada para a monitorização de parâmetros relacionados com saúde e bem estar do utilizador. Não se cingindo a nenhum público em específico, a instalação e uso dos serviços desta aplicação não apresenta nenhuma restrição de idade por parte dos utilizadores.

A aplicação MiFit está disponível para os principais sistemas operativos atualmente usados pelos dispositivos móveis, isto é, iOS e Android. Para o sistema Android, a aplicação requer Android 4.4 (Kitkat, API 19) ou superior, enquanto que no sistema proprietário da Apple, a versão mínima é iOS 10.0. Ambas as versões estão disponiveis nas maiores lojas de aplicações, no caso do Android na loja `Google Play Store`[1] e na `AppStore`[2] para o sistema iOS.

As funcionalidades disponibilizadas, fazem da MiFit uma aplicação com elevada popularidade na categoria de saúde e *fitness*. A Outubro de 2021, a aplicação conta mais de 2 000 000 de instalações a partir do no mercado de aplicações `Google Play Store` (Statista, 2021).

Tipicamente as aplicações para funcionarem corretamente requerem acesso a componentes e dados do SO, sendo que o utilizador vê-se obrigado a conceder autorizações à aplicação. As autorizações são disponibilizadas sobre forma de permissões, sendo esta uma forma eficaz de garantir a segurança e privacidade do utilizador.

O sistema Android assenta em centenas de permissões, permitindo que sejam acedidos componentes de software e hardware do dispositivo. Os avanços no hardware motivam fortemente tais evolução das permissões, removendo as obsoletas e que cubram as necessidades de segurança e privacidade dos utilizadores (Wei et al., 2012).

A aplicação MiFit não é exceção, requerendo nove permissões para o seu pleno funcionamento. A Tabela 3 lista as permissões solicitadas ao utilizador para o uso da aplicação MiFit.

1 https://play.google.com/store/apps/details?id=com.xiaomi.hm.health
2 https://apps.apple.com/us/app/mi-fit/id938688461

A recusa de alguma das permissões listadas podem limitar o acesso às funcionalidades da aplicação.

PERMISSÃO	DESCRIÇÃO
Armazenamento	Acesso a fotos e outros ficheiros do dispositivo.
Câmara	Captura de imagem ou gravação de vídeo pela câmara
Contactos	Acesso à lista de contactos
Localização	Solicitar a localização do dispositivo.
Mensagens	Ver e enviar mensagens de texto (SMS).
Microfone	Gravação de áudio
Registo de chamadas	Ver e alterar o histórico de chamadas
Sensores corporais	Acesso a informações de sensores sobre os sinais vitais.
Telefone	Fazer e gerir chamadas telefónicas.

Tabela 3: Lista de permissões da MiFit para Android

5.1 ANÁLISE ESTÁTICA

A análise estática da aplicação revelou-se um processo importante durante a realização deste projeto. Recorreu-se ao software Mobile Security Framework (MobSF)[3] para a realização deste tipo de análise estática automatizada da aplicação móvel MiFit. Tipicamente esta ferramenta analisa o código fonte sem executar a aplicação, no entanto, pode ser usada para testar código, caches ou outras vulnerabilidades deixadas durante o desenvolvimento (Shirke, 2019).

Seguidamente, são analisados vários itens com recurso à aplicação MobSF.

5.1.1 *Domínios dos serviços*

Foi levantada a lista de países que integram os domínios dos serviços da MiFit. O MobSF executa uma análise de *malware*[4] aos domínios apresentando um quadro de avaliação, facilitando assim a tarefa do perito forense. Os domínios estão maioritariamente localizados na China, Estados Unidos da América, Holanda, Singapura, Hong Kong, Alemanha, Espanha e Irlanda. Segundo o relatório, todos os domínios apresentam uma boa avaliação (*good*).

3 https://github.com/MobSF/Mobile-Security-Framework-MobSF
4 Malware é um software intrusivo que foi concebido para danificar e destruir sistemas informáticos.

5.1.2 *Análise de risco às permissões*

Com objetivo de estudar as permissões potencialmente perigosas que são requeridas pela aplicação, analisou-se o ficheiro `AndroidManifest.xml`.

A Tabela 4 lista as permissões identificadas como potencialmente perigosas pelo MobSF.

PERMISSÃO	DESCRIÇÃO
ACCESS_COARSE_LOCATION	Aceder a fontes de localização pouco exatas para determinar uma localização aproximada do telefone.
ACCESS_FINE_LOCATION	Aceder a fontes de localização exata, tais como GPS ao telefone.
ACCESS_BACKGROUND_LOCATION	Aacesso a fontes de localização, mesmo quando executada em segundo plano.
GET_TASKS	Obter informações sobre tarefas actualmente e recentemente em execução. Pode permitir que aplicações maliciosas descubram informação privada sobre outras aplicações.
WRITE_EXTERNAL_STORAGE	Escrita de conteúdos na zona de armazenamento externo.
READ_EXTERNAL_STORAGE	Ler conteúdos na zona de armazenamento externo.
READ_PHONE_STATE	Aceder às características do dispositivo, onde se inclue o número de série deste telefone, se uma chamada está ativa, o número a que está ligado entre outras informações.
CALL_PHONE	Efectuar chamadas telefónicas sem a intervenção do utilizador.
ANSWER_PHONE_CALLS	Responder a chamadas recebidas
GET_ACCOUNTS	Acesso à lista de contas armazenadas no serviço de contas (*Accounts Service*) do Android.
READ_CONTACTS	Ler todos os dados associados aos contactos armazenados no dispositivo.
READ_CALL_LOG	Ler o registo de chamadas.
REQUEST_INSTALL_PACKAGES	Permite à aplicação instalar *packages* adicionais. Aplicações maliciosas podem instalar conteúdo maliciosos no dispositivo sem o consentimento do utilizador.
ACTIVITY_RECOGNITION	Reconhecer e registar atividade física do utilizador.
RECORD_AUDIO	Gravação de áudio.
CAMERA	Acesso à câmara do dispositivo, incluindo captura de imagens e vídeo em qualquer instante.
SEND_SMS	Permite a aplicação de enviar SMS. Aplicações maliciosas podem enviar SMS sem o consentimento do utilizador.

Tabela 4: Permissões potencialmente perigosas na MiFit identificadas pelo MobSF

É do nosso crer que as permissões identificadas não tenham fins maliciosos ou mal intencionados, servindo apenas para disponibilizar as funcionalidades prometidas pela aplicação. A exceção é a permissão `RECORD_AUDIO`, dado que não identificado nos testes realizados nenhuma funcionalidade que requeira gravação de áudio. Contudo, não foi possível testar com todos os dispositivos passíveis de serem conectados à aplicação MiFit, podendo algum destes requerem essa permissão para proporcionar funcionalidades ao utilizador.

5.1.3 *Ligação a wearables*

Para o registo dos parâmetros relacionados com a saúde, a MiFit conecta-se a *wearables* externos, como relógios, pulseiras inteligentes, calçado, só para enumerar alguns. Dependendo do tipo de dispositivo que é conectado, os artefactos gerados são naturalmente diferentes.

A Figura 9 mostra o menu para adicionar um periférico/*wearable* à aplicação.

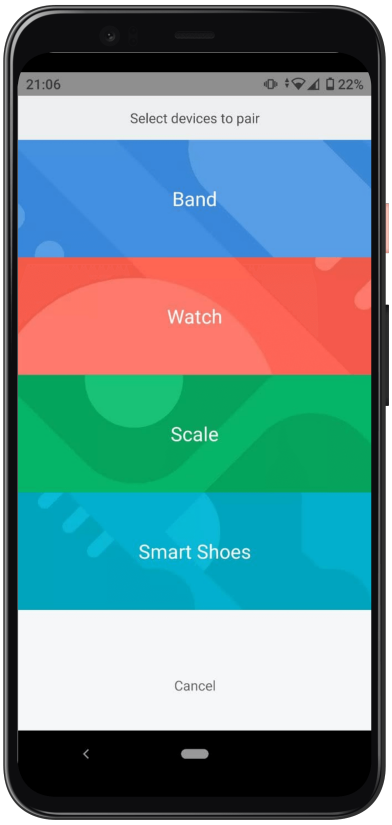

Figura 9: MiFit- Menu para adicionar periféricos

Para este projeto, recorreu-se à pulseira inteligente MiBand que foi alvo de escrutínio no Capitulo 4.

5.1.4 *Monitorização dos períodos de sono*

Como referenciado anteriormente, a MiBand regista vários parâmetros do utilizador, onde se incluem a monitorização dos períodos de sono. O separador referente à

monitorização da qualidade do sono, apresenta quatro níveis de granularidade: dia, semana, mês e ano.

Na granularidade mais baixa, a tabela de sono diária é apresentada uma linha temporal seccionada em listas verticais. As listas são identificadas com cores diferentes correspondentes às quatro fases do sono fase do sono. Ao manter premido uma das listas correspondentes a fase do sono, a aplicação mostra o intervalo temporal dessa fase.

A Figura 10 mostra o separador onde é possível observar os parâmetros do sono diário. Neste exemplo podemos observar que o utilizador para o dia 26 de Novembro de 2021 adormeceu perto das 00h57, despertou às 8h21, estando 4 minutos acordado entre as 02h55 e as 02h59.

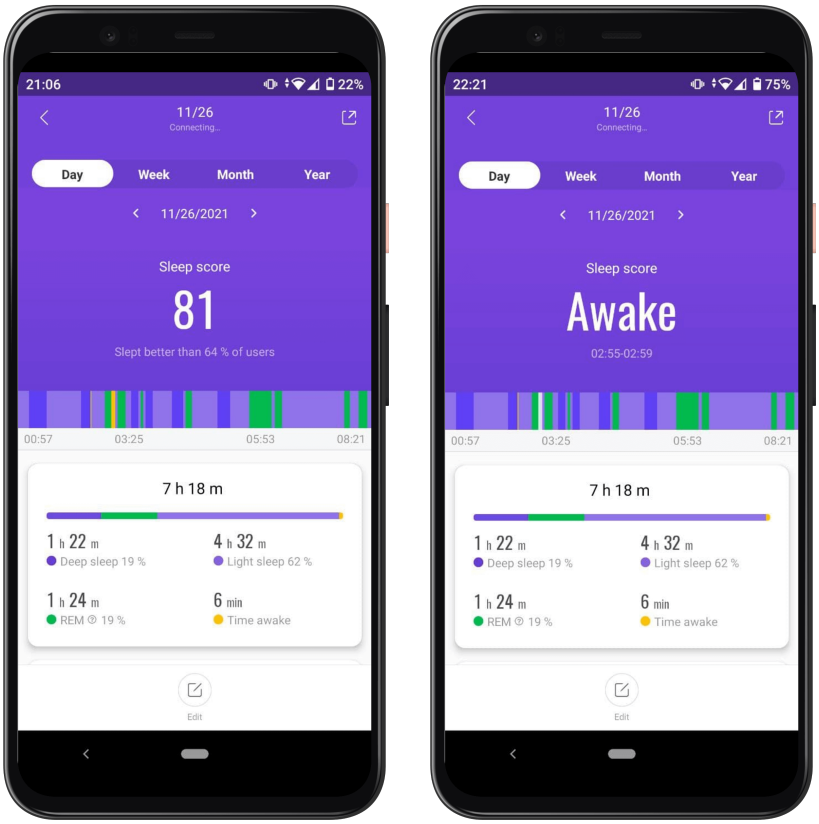

Figura 10: MiFit- Sono

Nas granularidades semana, mês e ano são apresentados os somatório estatísticos do tempo médio na fase sono profundo, sono leve, REM, tempo acordado, hora média do despertar e tempo médio acordado.

A Figura 11 mostra um exemplo do ecrã estatístico do sono para um mês.

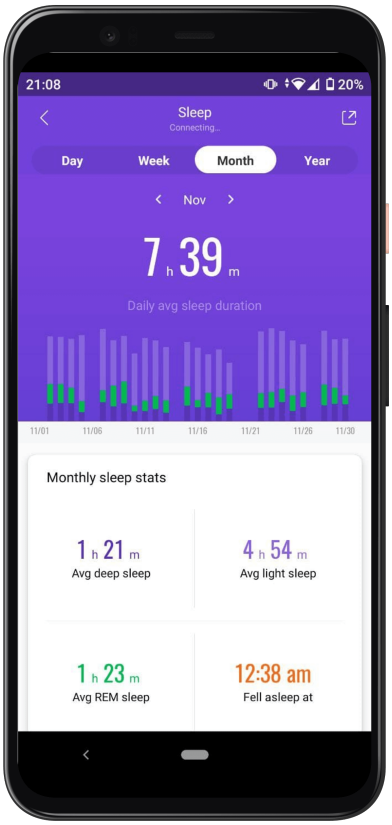

Figura 11: MiFit- Sono

Os parâmetros do sono, nomeadamente o período temporal em que a pessoa se encontra a dormir, podem revelar-se um importante artefacto forense, corroborando ou refutando hipóteses que podem ser cruciais num processo de investigação ao utilizador. Com a análise dos conteúdos da aplicação (mais adiante explanados) é possível perceber hábitos de sono e a sua qualidade.

5.1.5 *Monitorização de passos*

O ecrã da Figura 12 apresenta os dados capturados com a MiBand relativamente aos passos dados ao longo da atividade do utilizador.

A granularidade é diária, havendo a possibilidade de fazer *drill up*[5] para dados estáticos semanais, mensais e anuais, assemelhando-se ao ecrã dos parâmetros do sono descrito anteriormente.

5 Ver dados com menos detalhe, subindo na hierarquia para medidas com uma granularidade menos detalhada.

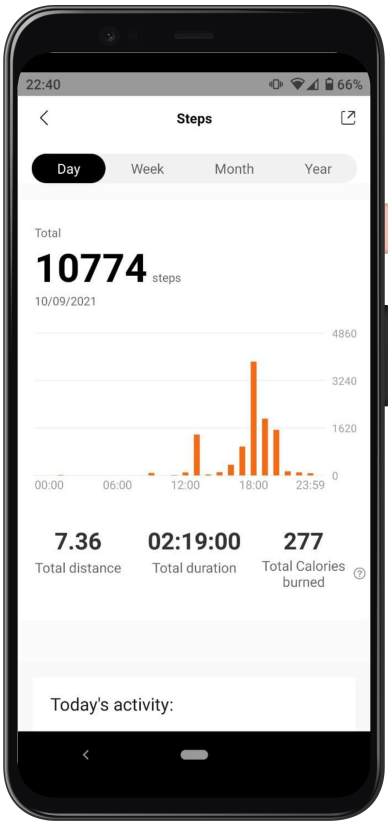

Figura 12: MiFit- Steps

Através de um algoritmo, que é do nosso crer que são baseadas no ritmo do batimento cardíaco e outras métricas, a aplicação MiFit identifica quando o utilizador inicia uma atividade desportiva ou altera o seu comportamento. Exemplo disso é o início de uma caminhada e respetivas oscilações de ritmo (início de uma caminhada lenta para caminhada rápida) e o início de uma atividade mais ligeira, só para enumerar algumas (Figura 13).

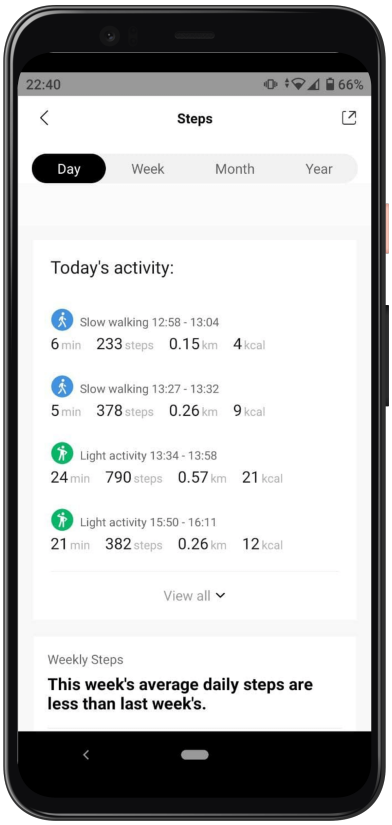

Figura 13: MiFit- Steps - Atividades Automáticas

5.1.6 *Stress*

Outra macro funcionalidade da MiFit é registar o stress do utilizador. Segundo a aplicação MiFit, o valor de stress medido é calculado pela combinação a variação do ritmo cardíaco e modelos de stress.

A medição do nível de stress pode ser efetuada de forma manual ou automática. Quando o utilizador opta por medir manualmente o nível de stress, a pulseira faz a medição durante aproximadamente 1 minuto. A sincronização dos dados entre a pulseira MiBand e a aplicação MiFit é feita em simultâneo com os outros dados capturados e sempre que a ligação Bluetooth esteja ativa entre a pulseira e o dispositivo gestor.

A Figura 14 representa o ecrã da aplicação MiFit no qual estão registados os valores de stress medidos com a pulseira MiBand.

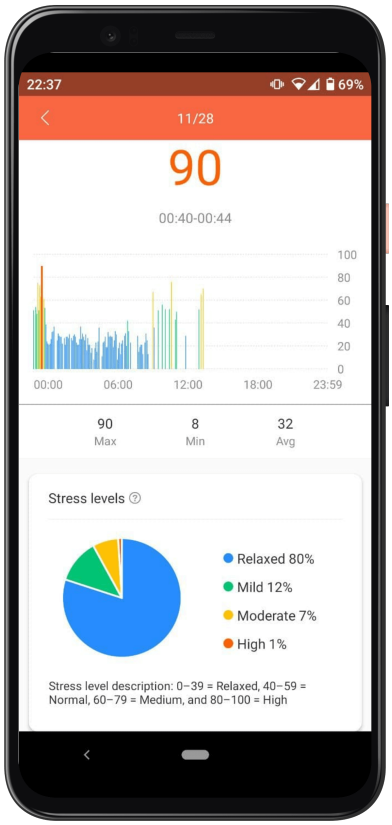

Figura 14: MiFit- Stress

Os valores estão compreendidos num intervalo de 0 a 100, onde 0 é o nível mais baixo e 100 representa o valor mais elevado, como mostra a Tabela 5 que contém a descrição dos níveis de stress divididos por intervalos e cor.

DESCRIÇÃO	INTERVALO	COR
Relaxado	0 a 39	Azul
Médio	40 a 59	Verde
Moderado	60 a 79	Amarelo
Alto	80 a 100	Vermelho

Tabela 5: Níveis de stress na MiFit

Caso o utilizador ative a opção automática, a pulseira MiBand faz medições de 5 em 5 minutos. Apesar dos valores medidos serem apenas de referência para o utilizador, a aplicação MiFit menciona que a taxa de sucesso é maior quando o utilizador está mais calmo.

Consoante a opção escolhida, automática ou manual, os dados são persistidos em zonas diferentes. As diferenças na persistência dos dados irão ser abordados no Capitulo 5.3.

5.1.7 *Ciclos*

O menu do ciclo de menstruação feminino apresenta uma interface muito completa na qual o utilizador deverá indicar os sintomas que apresenta durante o seu período e assim manter um registo de quaisquer alterações e facilitar um diagnóstico mais preciso sobre quais serão os dias da sua menstruação, período fértil e dias de ovulação.

Entre as opções para registar sintomas, existe a possibilidade de estabelecer o nível de dor (leve, moderada ou grave), quantidade de hemorragia (leve, média e elevada) e o humor (feliz, neutra e frustrada) (Figura 15).

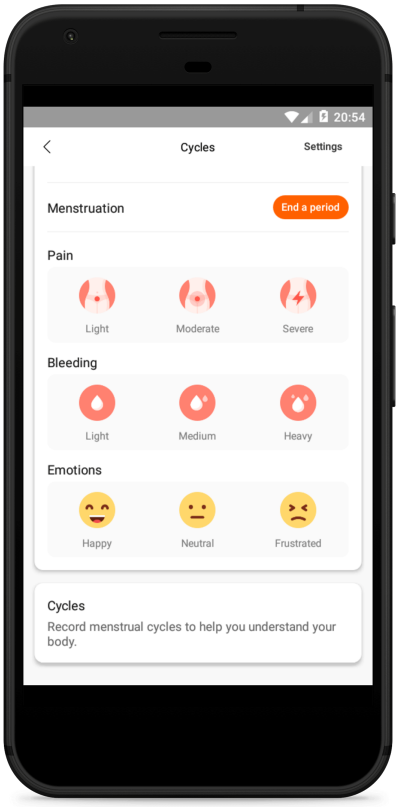

Figura 15: MiFit- Ciclos - Registo de sintomas

Tendo todos estes dados registados, é possível ver o registo diretamente no calendário presenta na aplicação MiFit ou na pulseira MiBand no menu `ciclos`.

A Figura 16 mostra os diferentes ecrãs da funcionalidade supracitada.

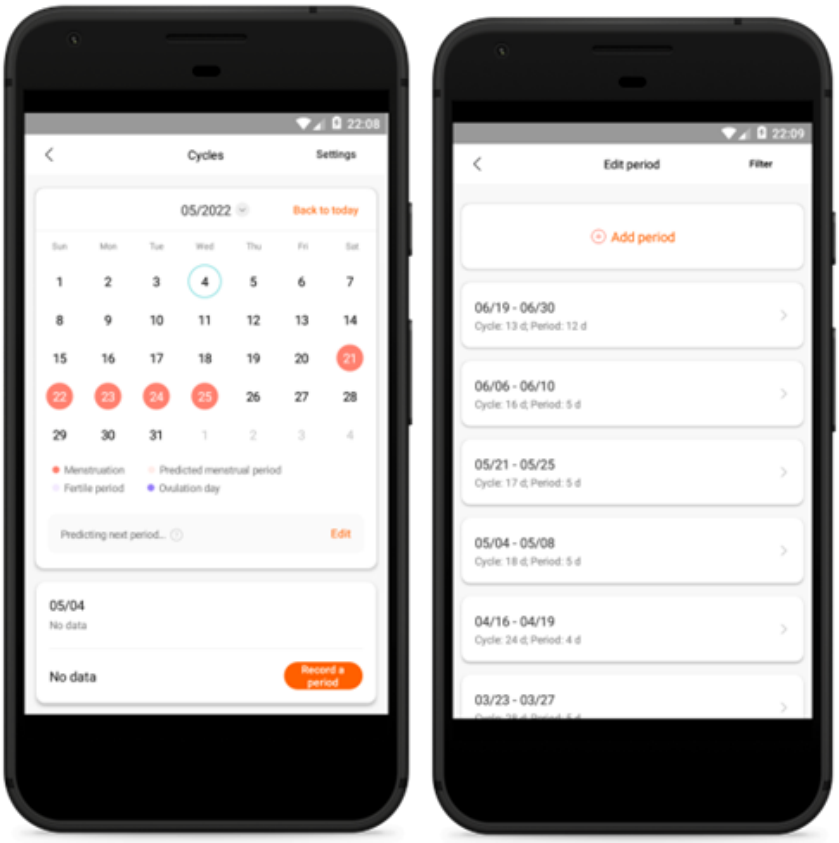

Figura 16: MiFit- Ciclos

5.2 ORGANIZAÇÃO INTERNA DA APLICAÇÃO MIFIT ANDROID

Quando a aplicação MiFit é instalada são extraídos e gerados os conteúdos necessários para o normal funcionamento da aplicação.

Analisando a aplicação do ponto de vista forense, é notório que os dados com maior relevância são mantidos na zona de armazenamento privada `com.xiaomi.hm.health`.

O acesso a esta diretoria requer privilégios de sistema. Para ultrapassar esta barreira imposta pelo sistema operativo, deve ser usado um utilizador com nível privilegiado, isto é, *root*. A Figura 17 mostra a hierarquia de diretorias da aplicação contidas em `com.xiaomi.hm.health`.

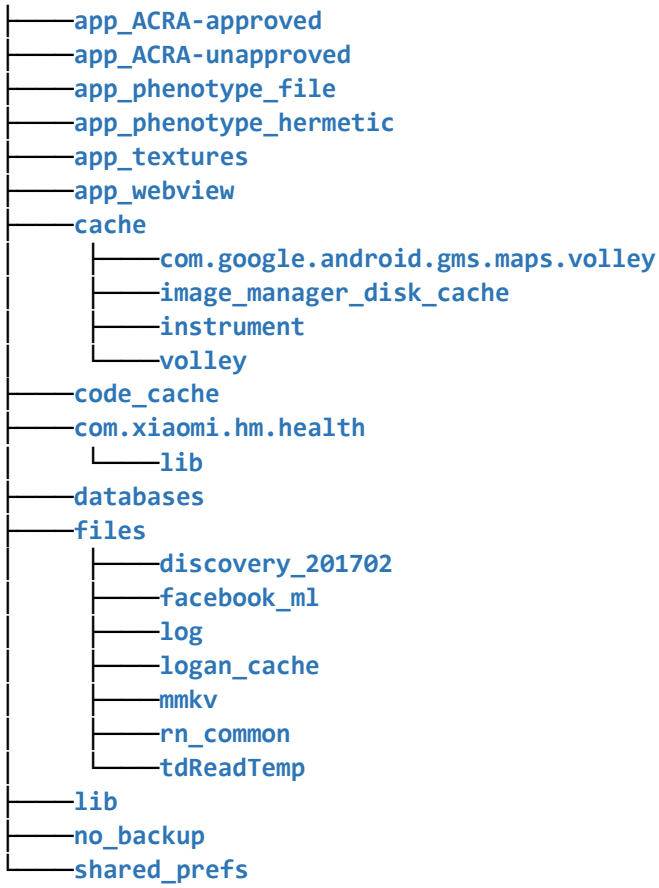

Figura 17: Hierarquia de diretorias da aplicação MiFit

As secções seguintes abordam a análise dos conteúdos gerados pela aplicação que fazem uso dos principais mecanismos de armazenamento disponíveis no sistema Android, onde se incluem as bases de dados da diretoria `databases` e os ficheiros `Shared Preferences` que existem na diretoria `shared_prefs`.

5.3 BASES DE DADOS

A aplicação MiFit para o sistema Android possui um total de 6 bases de dados SQLite3. Como observado na Figura 17, a diretoria `databases` consta na normal hierarquia das aplicações Android, sendo que todas as bases de dados, excepto uma - `Cookies` estão localizadas nesta diretoria. A base de dados `Cookies` está armazenada na diretoria `app_webview`. A Tabela 6 apresenta uma sucinta descrição das bases de dados da aplicação MiFit.

BASE DE DADOS	DESCRIÇÃO
Cookies	cookies de sessão da APP
FemaleHealth_<id>.db	informações do ciclo menstrual feminino
FitTime_<id>.db	(não foi possível descobrir a correta interpretação dos valores)
origin_db_<id>	informações do utilizador (incluindo sono, passos, atividades desportivas...)
spo2_<id>	informação do nível de saturação de oxigénio no sangue
stress_<id>.db	informação parcial de stress

Tabela 6: Bases de dados SQLite3 da aplicação MiFit para Android

As principais bases de dados que integram a aplicação seguem um padrão semelhante na sua nomenclatura. Ao ficheiro de base de dados é adicionado um sufixo correspondente à `hash` gerada pelo algoritmo Message Digest Algorithm (MD5).

A Listagem 2 mostra um exemplo concreto da aplicação do algoritmo MD5 no número identificador do utilizador `7598701544`, resultando no valor

`922775702a89350dd4dee25186f6068c`.

Seja:

1. `MD5`: a função de aplicação do algoritmo
2. `7598701544`: id do utilizador
3. `resultado`: valor obtido com a aplicação do algoritmo `MD5`

```
1    MD5(id) = resultado,
2    MD5(7598701544) = 922775702a89350dd4dee25186f6068c
```

Listagem 2: Aplicação do algoritmo MD5 ao identificador do utilizador

Podemos então concluir que o as informações guardadas na base de dados `origin_db_922775702a89350dd4dee25186f6068c` pertencem ao utilizador cujo o identificador é `7598701544`.

Foram identificadas falhas de segurança com diferentes níveis de severidade associadas ao algoritmo MD5, onde se inclui gerar uma colisão com diferentes entradas, vulnerabilidades em Public Key Infrastructures (PKI) e construção de certificados Secure Sockets Layer (SSL) falsos (Dougherty, 2009).

Neste contexto, uma colisão acontece quando dois valores de entrada distintos (ids) geram o mesmo valor de saída. Este comportamento não deve acontecer num algoritmo de *hashing*.

Mesmo que o histericamente o algoritmo MD5 seja considerado inseguro quando usado em funcionalidades criticas, é do nosso crer que o uso neste contexto não constitui qualquer falha de segurança que coloque em causa algum dos três principais pilares da segurança informática (confidencialidade, integridade e disponibilidade).

Analisaremos agora as bases de dados identificadas.

cookies. A base de dados `Cookies` possui duas tabelas: `cookies` e `meta`. A tabela `cookies` lista os *cookies*[6] de sessão usados durante as sessões aos recursos web da aplicação, realçando os atributos apresentados na Tabela 7.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
creation_utc	integer	instante temporal da criação do cookie
expires_utc	integer	instante temporal de expiração do cookie
last_access_utc	integer	(resultados obtidos não eram coerentes com o esperado)
value	text	valor (*string*) do cookie

Tabela 7: Principais atributos da tabela cookies da base de dados Cookies

A segunda tabela de base de dados identificada, `meta`, não apresenta valor forense, servindo apenas para controlo interno da aplicação.

femalehealth_<id>. A base de dados `FemaleHealth_<id>` regista informações do ciclo menstrual feminino. Do desenrolar dos testes efetuados à aplicação, foi usado um utilizador identificado como género masculino. Durante esse período, a base de dados `FemaleHealth_<id>` apresentou-se vazia.

Com o objetivo de extrair o máximo de conteúdos da aplicação, foi alterado o género na conta de utilizador, passando esta a assumir um utilizador feminino, possibilitando gerar dados usando a funcionalidade `cycles` mencionada anteriormente.

Para a versão da aplicação MiFit estudada, a base de dados `FemaleHealth_<id>` integra três tabelas: `MenstruationHistory`, `MenstruationRecord` e `Symptom`.

A tabela `MenstruationHistory` armazena as entradas dos dias de menstruação registadas manualmente. Os atributos `menstrualPeriod`, `menstrualEndTime`, `month` são parte integrante desta tabela e representam o dia inicial, o dia final e o mês que o utilizador definiu em formato Unix Timestamp. O atributo `userId`

6 Cookie é um termo que se refere a um pequeno pedaço de informação enviada para um website ou serviço, podendo conter informações do utilizador.

contém o número de utilizador que identifica inequivocamente o utilizador para qual a entrada foi submetida.

A tabela `Symptom` persiste os estados selecionados para o dor, quantidade de hemorragia e humor, como ilustrado na Figura 15.

A Tabela 8 lista os atributos considerados relevantes do ponto de vista forense da tabela de base de dados `Symptom`.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
date	integer	instante temporal do registo do sintoma
type	text	tipo do sintoma (MOOD, PAIN, BLOOD)
degree	integer	intensidade do sintoma

Tabela 8: Principais atributos da tabela Symptom da base de dados FemaleHealth_<id>

O atributo `type` assume um dos três valores: `MOOD` = humor, `PAIN` = dor, `BLOOD` = quantidade da hemorragia, correspondente ao tipo de sintoma registado. O atributo `date` armazena o dia para o qual foi submetido o sintoma.

Por fim, a tabela `MenstruationRecord` lista os dias de menstruação. O atributo `updateTime` contido na tabela representa a data em que foi registada a entrada na aplicação MiFit, complementado com o atributo `date` que representa o dia que a menstruação ocorreu. Ambos os atributos são representados no formato Unix Timestamp.

fittime_<id>. No decorrer dos testes efetuados a aplicação MiFit, a base de dados `FitTime\_<id>` apresentou-se vazia, impossibilitando a extração de conteúdo relevante para analise no âmbito deste projeto.

origin_db_<id>. É na base de dados `origin_db_<id>` que são persistidos grande parte dos dados relevantes do ponto de vista forense, contendo 61 tabelas, excluindo as tabelas de controlo `android_meta` e `sqlite_sequence`. Durante o decorrer a nossa análise, cerca de 54 tabelas mantiveram vazias, não tendo sido possível extrair qualquer informação relevante sobre o seu propósito ou qualquer valor forense. As tabelas populadas com base no uso da pulseira inteligente MiBand e a interação do utilizador e a aplicação MiFit são explanadas nos parágrafos seguintes.

HEART_RATE. A tabela `HEART_RATE` persiste os valores das medições dos ritmos cardíacos capturados pela MiBand. Cada linha da tabela corresponde a uma medição. O valor do batimento cardíaco medido pelo sensor é mapeado para o atributo `HR`, registando também o instante temporal que este foi aferido, persistindo-o no atributo `TIME` em formato UNIX Timestamp. Aquando o registo, é também persistido o identificador do dispositivo que fez a medição. Este identificador corresponde à chave estrangeira para a tabela `DEVICE` que será alvo de análise mais a frente neste documento. A Tabela 9 lista dos atributos considerados relevantes na tabela `HEART_RATE` da base de dados `origin_db_<id>`.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
TIME	integer	instante temporal da captura do batimento cardíaco
HR	integer	valor numérico do batimento cardíaca
DEVICE_ID	text	identificador do dispositivo de captura

Tabela 9: Principais atributos da tabela HEART_RATE da base de dados origin_db_<id>

ALARM. A funcionalidade de despertador presente na MiBand, quando usada gera dados dentro da aplicação MiFit. Os dados associados aos alarmes são persistidos na tabela `ALARM`. A Tabela 10 lista os principais atributos da tabela `ALARM` da base de dados `origin_db_<id>`.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
_id	integer	identificador sequencial numérico do alarme
CALENDAR	integer	(não foi possível descobrir a correta interpretação dos valores)
ENABLED	integer	0 = inativo, 1 = ativo

Tabela 10: Principais atributos da tabela ALARM da base de dados origin_db_<id>

DEVICE. Os dados armazenados na tabela `DEVICE` da base de dados `origin_db_<id>` permitem observar informações relevantes de dispositivos conectados à aplicação MiFit. Na lista de dispositivos passiveis de estabelecer conexão com a aplicação incluem-se *wearables* como a MiBand, a mesma que se tem revelado a principal produtor de dados armazenados no dispositivo no âmbito deste projeto.

Cada registo nesta tabela é mapeado para um *wearable* que o permite identificar inequivocamente como dispositivo que está em comunicação ou esteve conectado previamente. A Tabela 11 lista os atributos da tabela `DEVICE` considerados relevantes do ponto de vista forense.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
DEVICE_ID	text	identificador único
DEVICE_ADDRESS	text	endereço MAC
DEVICE_BIND_STATUS	integer	0=inativo, 1=ativo
DEVICE_BIND_TIME	integer	(resultados obtidos não foram coerentes com o esperado)
DEVICE_SYNC_DATA_TIME	integer	(resultados obtidos não foram coerentes com o esperado)
DEVICE_SYNC_DATA_TIME_HR	integer	(resultados obtidos não foram coerentes com o esperado)
AUTHKEY	text	(não foi possível descobrir a correta interpretação dos valores
SN	text	número de série
FIRMWARE_VERSION	text	versão do firmware

Tabela 11: Principais atributos da tabela DEVICE da base de dados origin_db_<id>

É de realçar que alguns dos identificadores/dados presentes na tabela `DEVICE` apenas são acessíveis através da base de dados, não sendo disponibilizados via App ou na pulseira MiBand.

USER_INFOS. A tabela `USER_INFOS` persiste informação dos utilizadores que se autenticaram na aplicação. Os atributos considerados relevantes do ponto de vista forense estão listados na Tabela 12.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
USER_ID	text	identificador numérico
NAME	text	nome
BIRTHDAY	text	data de nascimento
AVATAR_URL	text	endereço publico para a imagem de perfil
GENDER	integer	gênero, 0- feminino, 1 - masculino
HEIGHT	integer	altura
WEIGHT	real	peso
LONGITUDE	text	(não foi possível descobrir a correta interpretação dos valores)
LATITUDE	text	(não foi possível descobrir a correta interpretação dos valores)
CREATE_TIME	text	data de criação da conta
LAST_LOGIN_TIME	text	(resultados obtidos não foram coerentes com o esperado)

Tabela 12: Principais atributos da tabela USER_INFOS da base de dados origin_db_<id>

De realçar que estas informações podem não corresponder de forma exata ao utilizador, visto que dados como o nome, data de nascimento, altura e peso são inseridos manualmente pelo utilizador na aplicação.

Como referido anteriormente, a nomenclatura da base de dados `origin_db_<id>` onde a tabela `USER_INFOS` é parte integrante resulta da aplicação do algoritmo

criptográfico MD5, tendo como valor de entrada o valor armazenado no atributo `USER_ID`.

DATE_DATA. O sumário dos ciclos de sono e dos passos diário do utilizador são persistidos na tabela `DATE_DATA`, destacando os atributos `SOURCE`, `DATE` e `SUMMARY`, como listados na Tabela 13. A granularidade dos registos desta tabela são ao dia, já que cada entrada nesta tabela corresponde à atividade diária do utilizador.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
SOURCE	integer	identificador do dispositivo de captura
DATE	text	data
SUMMARY	text	resumo diário em formato json

Tabela 13: Principais atributos da tabela DATE_DATA da base de dados origin_db_<id>

O atributo `DATE` representa a data no formato aaaa-mm-dd. É no atributo `SUMMARY` que o resumo da atividade do utilizador é armazenado. Este resumo é armazenado num objeto complexo em formato JavaScript Object Notation (JSON).

Não é de estranhar que o tamanho de cada registo varia muito com o uso da pulseira MiBand, sendo que quanto mais variação no ritmo e tipo de atividades, maior será o objeto armazenado no atributo `SUMMARY`. Para melhor entender este complexo objeto, seccionaremos em duas propriedades: `slp` e `stp`, focando primeiramente na propriedade `slp`, parcialmente exemplificada na Listagem 3.

Esta propriedade é responsável por representar a atividade do sono monitorizada pela MiBand persistida na aplicação MiFit, constituindo assim um importante artefacto forense.

A Listagem 3 mostra parcialmente a propriedade `slp` no objeto `SUMMARY`.

```
{

   ...

   "slp":{
      "st":1631749800,
      "ed":1631775360,
      "dp":91,
      "lt":275,
      "wk":3,
      "usrSt":-1440,
```

```
        "usrEd":-1440,
        "wc":1,
        "is":44,
        "lb":59,
        "to":0,
        "dt":57,
        "rhr":60,
        "ss":86,
        "stage":[

          ...

           {
              "start":1490,
              "stop":1516,
              "mode":4
           },

           ...

        ]
    },

     ...

}
```

Listagem 3: Propriedade slp contida no objeto SUMMARY

Dentro da propriedade `slp` do objeto `SUMMARY` mencionado anteriormente, destacam-se seis atributos, explanados na Tabela 14.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
st	integer	inicio da atividade do sono
ed	integer	fim da atividade do sono
dp	integer	somatório da fase de sono profundo
lt	integer	somatório da fase de sono levo
wk	integer	somatório do tempo acordado
stage	array	lista de de intervalos de cada fase

Tabela 14: Atributos da propriedade slp do objeto SUMMARY

O atributo `st` marca o inicio da atividade do sono diário, enquanto que o atributo `ed` assinala o fim do período de sono diário. Ambos os atributos estão no formato UNIX Timestamp.

Os atributos `dp`, `lt` e `wk` representam o somatório dos tempos para cada uma das três fases do sono: `dp` a fase de *deep sleep* (sono profundo), `lt` a fase de *light*

sleep (sono leve) e `wk` para *time awake*, ou seja, o tempo em que o utilizador esteve acordado durante o dia em análise.

A lista de intervalos onde são registadas as diferentes fases de sono são mantidos no atributo `stage`. Cada objeto contido na lista `stage` corresponde a uma fase distinta, sendo que dois ciclos adjacentes terão necessariamente fase de sono distintos. Programaticamente, `stage` é uma lista de objetos em formato JSON composto por três atributos: `start`, `stop` e `mode`.

O atributo `start` é o número de segundos após o inicio da atividade do sono (`st`) e marca o inicio de uma nova fase do sono.

O atributo `stop` é o número de segundos após o inicio da atividade do sono (`st`) e marca o fim de uma fase do sono.

Existem quatro tipos de fases distintas `Sono Leve`, `Sono Profundo`, `Tempo Acordado` e `REM` a serem atribuídos ao atributo `mode`. Cada tipo fase de sono na aplicação MiFit é representado por um número inteiro. A Tabela 15 mapeia identificador numérico atribuído pela aplicação para o tipo de fase de sono apresentado ao utilizador na MiFit.

MODO	DESCRIÇÃO
4	Sono Leve (Light Sleep)
5	Sono Profundo (Deep Sleep)
7	Tempo acordado (Time Awake)
8	REM

Tabela 15: Tipos de fases de sono na MiFit

Sintetizando o exemplo da Listagem 3, o utilizador iniciou a sua atividade de sono do dia 15 Setembro de 2021 às 23:50:00 e terminou no dia 16 de Setembro de 2021 pelas 06:56:00. Durante esse período, esteve 91 minutos em sono profundo, 275 minutos em sono leve e 3 minutos acordado. No objeto contido no atributo `stage`, constatamos ainda que o utilizador este acordado durante 26 segundos entre as 00h:14m:50s e as 00h:15m:16s do dia 16 de Setembro de 2021.

Analisaremos agora a propriedade `stp` do objeto `SUMMARY`. É nesta propriedade que são armazenadas informações dos passos dados diariamente pelo utilizador da MiBand. A Listagem 4 representa parte da propriedade `stp` que é parte integrante do objeto `SUMMARY`.

```
1 {
2
```

```
3       ...
4
5     "stp":{
6        "ttl":7119,
7        "dis":5235,
8        "cal":219,
9        "wk":97,
10       "rn":0,
11       "runDist":204,
12       "runCal":32,
13       "stage":[
14
15          ...
16
17          {
18             "start":518,
19             "stop":589,
20             "mode":7,
21             "dis":460,
22             "cal":17,
23             "step":590
24          },
25
26          ...
27
28       ]
29    }
30
31    ...
32
33 }
```

Listagem 4: Propriedade stp contida no objeto SUMMARY

Ao contrário da propriedade `slp` mencionada anteriormente, a propriedade `stp` não possui nenhum atributo indicativo do instante temporal inicial e final, visto que a contagem diária tem sempre inicio as 00h:00m:00s do dia e termina às 23h:59m:59s.

De relembrar que o objeto `SUMMARY` onde está inserida a propriedade `stp` é parte integrante de uma entrada na tabela de base de dados `DATE_DATA`, contendo o atributo `DATE` que indica a data para a qual o objeto `SUMMARY` se refere. Na Tabela 16 são descritos os oito atributos que compõem a propriedade `stp`.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
ttl	integer	total de passos
dis	integer	distância em metros
cal	integer	calorias consumidas
wk	integer	tempo em minutos a andar
rn	integer	tempo em minutos a correr
runDist	integer	distância (corrida)
runCal	integer	calorias consumidas (corrida)
stage	array	lista de intervalos de cada atividade

Tabela 16: Atributos da propriedade stp do objeto SUMMARY

O atributo `ttl` revela o número total diário de passos dados percorrendo uma distância total refletida no atributo `dis`. O atributo `cal` indica as calorias consumidas. O atributo `wk` e `rn` representam o tempo em minutos a andar e a correr respetivamente. O atributo `runDist` representa o somatório da distancia percorrida em corrida, enquanto `runCal` as calorias consumidas durante esse período.

O atributo `stage` da propriedade `stp` apresenta uma estrutura semelhante ao atributo com o mesmo nome na propriedade `slp`. Programaticamente, é uma lista de objetos representativos de pequenos intervalos de atividades. A Tabela 17 descreve os seis atributos que compõem os objetos contidos no atributo `stage` da propriedade `stp`.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
start	integer	minuto do início da atividade
stop	integer	minuto do fim da atividade
mode	integer	tipo de atividade
dis	integer	distãncia percorrida
cal	integer	calorias consumidas
step	integer	passos percorridos

Tabela 17: Atributos do objeto contido na lista stage no objeto stp

O atributo `start` representado por um número inteiro positivo, corresponde ao minuto do início da atividade a partir das 00h:00m:00s. Seguindo a mesma estrutura e tipo de dados que o anterior, o atributo `stop` representa o minuto do fim da atividade.

Durante o desenvolvimento desde projeto, foram identificadas quatro tipos de atividades mapeadas para um identificador numérico persistido no atributo `mode`.

A Tabela 18 mostra os modos das atividades desportivas identificadas associados ao identificador numérico.

MODO	DESCRIÇÃO
1	caminhada lenta
3	caminhada rápida
4	corrida
7	atividade ligeira

Tabela 18: Modos de atividades na MiFit

A distancia percorrida entre os instantes `start` e `stop` na atividade `mode` é dada pelo atributo `dis`, associada as calorias consumidas (`cal`) e ao número de passos dados (`step`).

Com este importante artefacto forense é possível traçar parte do perfil de um utilizador e perceber se num determinado instante este se encontrava em movimento e o ritmo do mesmo, sendo esse último percetível pelo atributo `mode`.

Sintetizando o exemplo da Listagem 4, o utilizador no dia DATE[7] percorreu uma distância (`dis`) de 5235 metros em 7119 passos (`ttl`) num tempo estimado (`wk`) de 01h:37m (97 minutos) consumindo 219 quilocalorias (`cal`). Durante 71 minutos, entre as 08h:38m e as 09h:49m esteve numa atividade considerada ligeira, percorrendo uma distância de 460 metros com 590 passos consumindo 17 quilocalorias nesse período.

TRACKRECORD. A tabela `TRACKRECORD` da base de dados `origin_db_<id>` regista as atividades desportivas (*workouts*) despoletadas manualmente pelo utilizador. No decurso dos testes efetuados para versão da aplicação MiFit estudada, a tabela apresenta cerca de 61 atributos, dos quais foram destacados oito. Foram selecionados os atributos que se apresentaram fortemente populados após os testes efetuados à aplicação MiFit, por apresentarem um potencial valor superior do ponto de vista forense.

A Tabela 19 lista os atributos selecionados da tabela de base de dados `TRACKRECORD`.

7 Data não visível na Listagem 4 referente ao atributo SUMMARY. Esta informação é armazenada no atributo `DATE` da tabela de base de dados `DATE_DATA`

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
DATE	text	data da atividade
TYPE	integer	tipo de atividade
TRACKID	integer	instante temporal do inicio da atividade
ENDTIME	integer	instante temporal do fim da atividade
DISTANCE	integer	distancia percorrida
CAL	text	calorias consumidas
AVGHR	integer	média do batimento cardíaco
LOCATION	text	(não foi possível descobrir a correta interpretação dos valores)
TOTAL_STEP	integer	total de passos

Tabela 19: Principais atributos da tabela TRACKRECORD da base de dados origin_db_<id>

O atributo `DATE` revela a data da atividade no formato `aaaa-mm-dd`.

O tipo de atividade é dado pelo atributo `TYPE`. O tipo da atividade é identificado numericamente, no entanto, este atributo apresenta um mapeamento distinto do apresentado para a tabela `DATE_DATA` (Tabela 18).

A Tabela 20 mostra o mapeamento entre as atividades desportivas e o identificador numérico persistido na base de dados.

MODO	DESCRIÇÃO	MODO	DESCRIÇÃO
8	passadeira	60	ioga
10	ciclismo *indoor*	61	pilates
12	elíptica	74	dança de rua
14	natação em piscina	76	dança
16	livre	77	zumba
21	saltar à corda	78	cricket
23	máquina de remo	80	bowling
24	treino *indoor*	85	basquetebol
45	patinagem no gelo *indoor*	88	voleibol
49	HIIT	89	ténis de mesa
50	treino de *core*	92	badminton
53	alongamento	97	boxe
58	*stepper*	104	kickboxing
59	ginástica		

Tabela 20: Identificadores das atividades na MiFit

O atributo `TRACKID` representa o instante temporal do inicio da atividade em formato Unix Timestamp. Este atributo identifica inequivocamente a atividade, servindo de chave para outras tabelas de base de dados.

A distância percorrida durante a atividade é dada pelo atributo `DISTANCE`, as quilocalorias consumidas pelo atributo `CAL` e o atributo `AVGHR` é um valor inteiro numérico representativo da média do ritmo dos batimentos cardíacos durante o período de atividade.

O `TOTAL_STEP` é o número de passos dados entre o instante temporal marcado pelo `TRACKID` e `ENDTIME`. A duração da atividade em segundos é obtida com a diferença ente estes dois instantes temporais.

TRACKDATA. A tabela de base de dados `TRACKDATA` está diretamente relacionada com a tabela `TRACKRECORD` descrita anteriormente, fornecendo informações mais detalhadas sobre as atividades desportivas. Esta relação é estabelecida através do atributo `TRACKID` que é chave estrangeira[8] para o atributo com o mesmo nome na tabela `TRACKRECORD`.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
TRACKID	integer	identificador da atividade
BULKLL	text	coordenadas GPS da atividade

Tabela 21: Principais atributos da tabela TRACKDATA da base de dados origin_db_<id>

O atributo `BULKLL` armazena as coordenadas GPS na forma de lista ordenada. Cada elemento da lista é separado com o símbolo `;` (ponto e virgula) contendo a latitude e longitude separado pelo símbolo `,` (virgula).

A Listagem 5 ilustra um exemplo dos dados armazenados no atributo `BULKLL`.

```
1  400000031,-88784775;-799,-967;-1534,267;-2299,-2900;-1165,-1166;-1634,-632;-1
   ↪ 732,-400;-1432,-799;-1667,-166;-1533,-233;-1366,33;-1132,333;-1067,400;-1
   ↪ 033,365;-1067,400;
2  ...
```

Listagem 5: Exemplo parcial do atributo BULKLL da tabela TRACKDATA

A coordenada consecutiva é calculada com base na sua antecessora. Para o exemplo supracitado, o utilizador iniciou o percurso na coordenada `40.0000031,-8.8784775`, seguindo para a coordenada `39.9999232, -8.8785574`, calculada pela soma entre ambos os valores representativos da latitude(-799) e longitude(-967).

8 Chave estrangeira é um ou mais atributos de uma tabela que se refere à chave primária de outra tabela, estabelecendo uma ligação entre ambas.

spo2_<id>. A base de dados spo2_<id> integra 7 tabelas de bases de dados (excluindo as tabelas de controlo do Android) sendo que apenas a tabela click_measured_spo2 aparenta ter relevância do ponto de vista forense. A Tabela 22 lista os principais atributos da tabela click_measured_spo2 da base de dados spo2_<id>.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
userId	integer	identificador numérico do utilizador
utcTimestamp	text	instante temporal da captura do spo2
spo2	integer	calor numérico do spo2
deviceId	text	identificador do dispositivo de captura

Tabela 22: Principais atributos da tabela click_measured_spo2 da base de dados stress_<id>

O atributo userId determina o utilizador que efetuou a medição de saturação do oxigenio (spo2) no instante temporal utcTimestamp no formato Unix Timestamp.

O atributo deviceId identifica o dispositivo de medição (MiBand), semelhante ao que acontece com outras tabelas responsáveis por registar medições de parâmetros relacionados com a saúde e bem-estar do utilizador.

O atributo spo2 persiste o valor medido pelo dispositivo deviceId. O dispositivo de captura sobre o qual foram realizados os testes é a MiBand 6, podendo registar medições num intervalo de 80% a 100% como indicado na aplicação MiFit.

stress_<id>. A base de dados Stress_<id> regista as medições automáticas e manuais do nível de stress. Com os dados populados nesta tabela é possível identificar momentos de stress sentidos pelo utilizador. Para cada medição do valor de stress está associado a informação temporal.

A Tabela 23 descreve os principais atributos da tabela AllDayStress da base de dados stress_<id>.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
timestamp	integer	instante temporal do registo (UNIX Timestamp milissegundos)
userId	text	identificador do utilizador
deviceId	text	identificador do dispositivo de captura
deviceMac	text	MAC Address do dispositivo de captura
deviceSn	text	número de serie do dispositivo de captura
maxStress	integer	valor máximo de stress diário
minStress	integer	valor mínimo de stress diário
avgStress	integer	média dos valores de stress diário
data	text	lista de medições de stress

Tabela 23: Principais atributos da tabela AllDayStress da base de dados stress___<id>

O atributo `timestamp` representado em formato Unix Timestamp em milissegundos regista o instante temporal em que foi feita a medição. O atributo `userId` funciona como chave estrangeira para o atributo `USER_ID` da tabela `USER_INFOS` da base de dados `origin_db_<id>` referente ao utilizador que fez a medição.

À semelhança do que acontece na tabela `DEVICE` da base de dados `origin_db_<id>`, os atributos `deviceId`, `deviceMac` e `deviceSn` fornecem informações que permitem identificar inequivocamente o acessório ligado à aplicação MiFit que foi responsável pela medição dos níveis de stress do utilizador.

Os atributos `minStress`, `maxStress` e `avgStress` correspondem ao valor mínimo, máximo e média respetivamente dos valores obtidos de stress. Estes valores são mostrados no separador de stress da aplicação MiFit como referido na Figura 14 a quando descrição das funcionalidades da aplicação.

O atributo `data` contém uma lista de medições de stress em formato JSON. Cada elemento nessa lista é representado por um objeto simples representativo de uma medição onde `time` reflete o instante temporal em formato Unix Timestamp e `value` o valor de stress calculado.

A Listagem 6 é parte da lista com os objetos contidos no atributo `data`.

Listagem 6: Lista de objetos contidos no atributo data

```
[

    ...

  {
    "time": 1636205640000,
    "value": 67
  },
  {
```

```
    "time": 1636206240000,
    "value": 46
  },
  {
    "time": 1636209840000,
    "value": 59
  },
  {
    "time": 1636210740000,
    "value": 52
  },
  {
    "time": 1636216740000,
    "value": 59
  },
  {
    "time": 1636219740000,
    "value": 54
  },

    ...

]
```

Como referido anteriormente, existe a possibilidade de registar o stress de forma não automática. Os registos feitos desta forma são persistidos na tabela `SingleStress`. Os atributos da tabela `SingleStress` assemelham-se aos que compõem a tabela supracitada `AllDayStress`. A Tabela 24 lista os atributos considerados relevantes.

ATRIBUTO	TIPO DE DADOS	DESCRIÇÃO
timestamp	integer	instante temporal do registo
userId	text	identificador do utilizador
value	integer	valor de stress medido
deviceId	text	identificador do dispositivo de captura
deviceMac	text	MAC Address do dispositivo de captura
deviceSn	text	número de serie do dispositivo de captura

Tabela 24: Principais atributos da tabela SingleStress da base de dados stress___<id>

Na tabela `SingleStress` destaca-se o atributo `value` que corresponde ao valor numérico de stress lido pela pulseira MiBand.

Os restantes atributos desempenham uma função análoga aos atributos com o mesmo nome da tabela `AllDayStress`.

5.4 SHARED PREFERENCES

Analisaremos agora a `Shared Preferences` da aplicação MiFit.

O ficheiro `hm_id_sdk_android.xml` inclui algumas informações do utilizador e de serviços da aplicação.

A Listagem 7 exibe o ficheiro `hm_id_sdk_android.xml`. De notar que parte do conteúdo do ficheiro foi censurado por questões de privacidade.

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
    <map>
        <string name="ti">
        {
            "domain": {
                "id-dns": "https://account-de2.huami.com"
            },
            "domains": [
                {
                    "cnames": [
                        "account-de2.huami.com"
                    ],
                    "host": "account.huami.com",
                    "ttl": 0
                },
                {
                    "cnames": [
                        "api-user-de2.huami.com"
                    ],
                    "host": "api-user.huami.com",
                    "ttl": 0
                },
                {
                    "cnames": [
                        "api-mifit-de2.huami.com"
                    ],
                    "host": "api-mifit.huami.com",
                    "ttl": 0
                },
                {
                    "cnames": [
                        "api-watch-de2.huami.com"
                    ],
                    "host": "api-watch.huami.com",
                    "ttl": 0
                },
                {
                    "cnames": [
                        "auth-de2.huami.com"
                    ],
```

```
                    "host": "auth.huami.com",
                    "ttl": 0
                },
                {
                    "cnames": [
                        "api-analytics-de.huami.com"
                    ],
                    "host": "api-analytics.huami.com",
                    "ttl": 0
                },
                {
                    "cnames": [
                        "app-analytics-de.huami.com"
                    ],
                    "host": "app-analytics.huami.com",
                    "ttl": 0
                }
            ],
            "provider": "google",
            "regist_info": {
                "country_code": "PT",
                "is_new_user": 0,
                "region": "10",
                "regist_date": 1628288588187
            },
            "token_info": {
                "app_token": "<APP_TOKEN>",
                "app_ttl": 43200,
                "lu_app_ttl": 1636302856376,
                "lu_ttl": 1636302856376,
                "login_token": "<LOGIN_TOKEN>",
                "ttl": 31536000,
                "mutime_long": 0,
                "user_id": "<USERID>"
            },
            "thirdparty_info": {
                "icon": "https://lh3.googleusercontent.com/a/AATXAJzZ5lS1nLxLSg_⌋
                ↪  862HtsQyLuY4Yeh_P86HlPSdf\u003ds96-c",
                "email": "<EMAIL>",
                "expires_in": 0,
                "nickname": "<NICKNAME>",
                "third_id": "111747675002256626568"
            },
            "mutime_long": 0,
            "result": "ok"
        }
        </string>
    </map>
```

Listagem 7: Conteúdo parcial do ficheiro hm_id_sdk_android.xml

A informação do atributo `domains` engloba uma lista de domínios e respectivos Canonical Name (CNAME)[9] usados para aceder aos serviços da aplicação.

O objeto contido `provider` contém o método de autenticação escolhido no início da última sessão na aplicação MiFit. No exemplo (Listagem 7) o provedor do serviço de autenticação é o Google, cujo as informações do utilizador autenticado são pormenorizadas no atributo `thirdparty_info`. O atributo `thirdparty_info` inclui o `email`, `nome de utilizador` e `identificador unico`, informações que permitem ao perito forense explorar outras contas em provedores de micro serviços externos.

O atributo `token_info` contempla conhecimento para o acesso aos serviços supracitados, onde se incluem *token* de autenticação e *token* de acesso aos serviços da aplicação. Tipicamente estes *tokens* os possuem um Time to live (TTL) curto, isto é, o tempo de validade é reduzido.

Foram realizados vários estudos para testar os *tokens* encontrados no ficheiro `hm_id_sdk_android.xml` no atributo `token_info`. O processo de testagem de acesso à aplicação passou por desenvolver um `script` Python capaz de estabelecer uma ligação à API da MiFit (`https://api-mifit-de2.huami.com`) recorrendo à biblioteca requests[10]. O *script* pede o recurso `history.json` do servidor, guardando a resposta no ficheiro `result.json`. De realçar que este método só funciona enquanto o *token* estiver válido.

O código do *script* pode ser consultado na Listagem 8.

```
1  import requests
2
3  APP_TOKEN = 'DQVBQFJyQktGHlp6QkpbRl5LRl5qek4uXAQABAAAAAECRX_KN2uSjHue7bmO ...'
4
5  result =
   ↪  requests.get('https://api-mifit-de2.huami.com/v1/sport/run/history.json',
   ↪  headers={'apptoken': APP_TOKEN}, params={'source': 'run.mifit.huami.com'})
6
7  f = open("result.json", "w")
8  f.write(str(result.json()))
9  f.close()
10
11 print("Done")
```

Listagem 8: Script para obter dados das atividades por API

9 Registo CNAME é um tipo de registo Domain Name System (DNS) que mapeia um nome para um nome de domínio.
10 https://pypi.org/project/requests/

A Listagem 9 apresenta o resultado parcial do pedido à API da MiFit e guardado no ficheiro `result.json`.


```
{
"code": 1,
"message": "success",
"data": {
    "next": -1,
    "summary": [

        ...

        {
            "trackid": "1650116696",
            "source": "run.212.huami.com",
            "dis": "24513.0",
            "calorie": "1090.0",
            "end_time": "1650126064",
            "run_time": "8658",
            "avg_pace": "0.353",
            "avg_frequency": "0.0",
            "avg_heart_rate": "115.0",
            "type": 9,
            "location": "ez19f5uy5jrg",
            "city": "",
            "forefoot_ratio": "-1",
            "bind_device": "0:MILI_PANGU_L: 212:V0.82.17.3",
            "max_pace": 177.0,
            "min_pace": 0.17764379,
            "version": 12,
            "altitude_ascend": 119,
            "altitude_descend": 72,
            "total_step": 60,
            "avg_stride_length": 0,
            "max_frequency": 0,
            "max_altitude": -20000,
            "min_altitude": -20000,
            "lap_distance": -1,
            "sync_to": "",
            "distance_ascend": 1918,
            "max_cadence": -1,
            "avg_cadence": -1,
            "landing_time": -1,
            "flight_ratio": -1,
            "climb_dis_descend": 1031,
            "climb_dis_ascend_time": 631,
            "climb_dis_descend_time": 433,
            "child_list": "",
            "parent_trackid": -1,
            "max_heart_rate": 172,
            "min_heart_rate": -1,
```

```
                "swolf": -1,
                "total_strokes": -1,
                "total_trips": -1,
                "avg_stroke_speed": -1.0,
                "max_stroke_speed": -1.0,
                "avg_distance_per_stroke": -1.0,
                "swim_pool_length": -1,
                "te": -1,
                "swim_style": -1,
                "unit": -1,
                "add_info": "",
                "sport_mode": 0,
                "downhill_num": 0,
                "downhill_max_altitude_desend": 0,
                "fore_hand": 0,
                "back_hand": 0,
                "serve": 0,
                "second_half_start_time": 0,
                "rope_skipping_count": 0,
                "rope_skipping_avg_frequency": 0,
                "rope_skipping_max_frequency": 0,
                "rope_skipping_rest_time": 0,
                "left_landing_time": -1,
                "left_flight_ratio": -1,
                "right_landing_time": -1,
                "right_flight_ratio": -1,
                "marathon": "",
                "situps": 0,
                "anaerobic_te": -1,
                "target_type": 0,
                "target_value": "0",
                "total_group": 0,
                "auto_recognition": false,
                "app_name": "com.xiaomi.hm.health",
                "heartrate_setting_type": 0,
                "heart_range": "780, -1;3960,-1;3120,-1;600,-1;180,-1;0,-1"
            },

            ...

        ]
    }
}
```


Listagem 9: Resultado do script para obter dados das atividades por API

Verificou-se que os atributos do objeto recebido no pedido à API assemelha-se aos campos encontrados na tabela de base de dados responsável pela persistência das atividades do utilizador.

O atributo `regist_info` armazena informação do registo de conta nos serviços da MiFit. De relembrar que o uso das funcionalidades da MiFit pressupõe que seja

criada uma conta de utilizador, independentemente do método de autenticação escolhido. É ainda armazenado o país de registo (`country_code`) e o instante temporal da data de criação de conta (`regist_date`) em formato Unix Timestamp em milissegundos.

5.5 SUMÁRIO

Este capítulo descreveu a estrutura da aplicação MiFit para o sistema Android. Foram ainda abordadas as fontes de dados presentes na aplicação, como as bases de dados SQLite3 e *shared_preferences*, com especial interesse nos conteúdos que apresentado valor do ponto de vista forense. O capítulo seguinte aborda o software de análise forense Autopsy.

6

AUTOPSY

Para assegurar as necessidades de uma análise forense digital e tornar mais célere o trabalho do perito forense, a comunidade interessada no tema tem desenvolvido varias soluções, algumas comerciais, outras em regime de código fonte aberto (*open-source*).

O Autopsy[1] é uma ferramenta em código fonte aberto que simplifica o processo de análise forense. Usado essencialmente em análises *post-mortem*, os resultados são apresentados em formato de artefactos.

A ferramenta Autopsy teve um papel de relevo durante o desenvolvimento deste projeto, pelo facto da plataforma de análise desenvolvida (MiFitAnalyzer, Capítulo 7) poder ser executada neste ambiente.

O presente capítulo enquadra o software Autopsy no contexto deste projeto, apresentado as noções essenciais para o seu uso, expandindo as suas funcionalidades recorrendo a módulos programáveis.

6.1 TIPOS DE MÓDULOS

As peculiaridades de cada investigação elevam a importância da temática da extensibilidade das funcionalidades do software Autopsy. Essa extensibilidade de funcionalidades é conseguida com a programação de módulos externos desenvolvidos em `Java` e/ou `Python`. Para melhor compreender o tipo de módulos que podem ser integrados na ferramenta de análise forense, é importante conhecer o seu ciclo de vida.

A Figura 18 ilustra o ciclo de vida de uma execução.

1 https://www.autopsy.com/

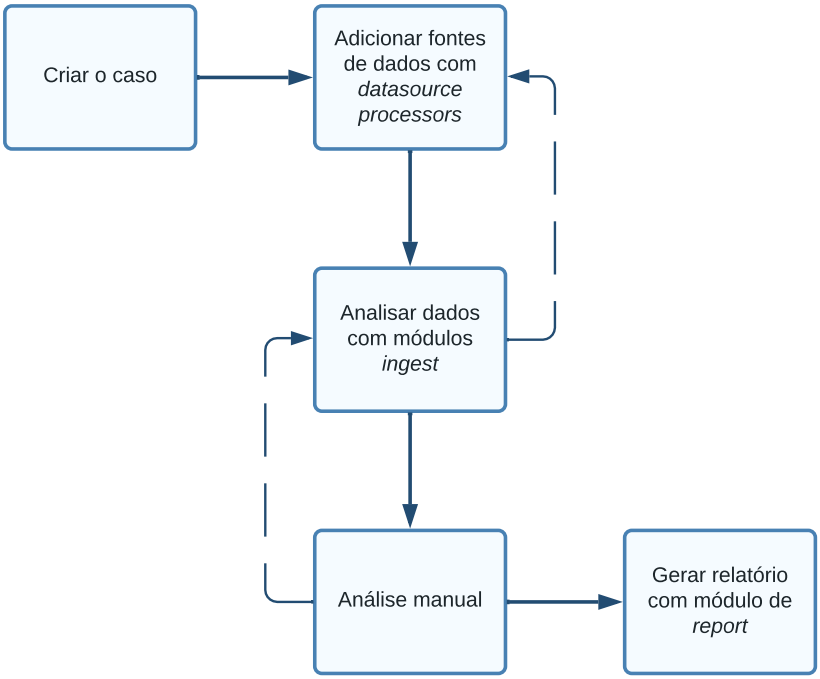

Figura 18: Ciclo de vida de uma execução no Autopsy

A Listagem seguinte sumariza os passos de uma execução dita normal no Autopsy.

1. Inicialmente, o perito forense começa por fornecer alguns dados e atributos capaz de identificar o caso em curso.
2. Fornecidos os dados do caso, são listados os módulos de `DataSource Processors` (Secção 6.1.1) responsáveis por definir a metodologia para adicionar as fontes de dados.
3. Escolhido o módulo de `DataSource Processors` adequado, o perito forense adiciona os ficheiros para análise como fontes de dados.
4. Adicionadas as fontes de dados, são escolhidos os módulos de `Ingest`. Os módulos de `Ingest` são responsáveis por processar as fontes de dados, identificando e indexando os artefactos forenses encontrados.
5. Terminado o processamento de todas as fontes de dados usando os módulos de `ingest`, é visível a árvore de artefactos encontrados, onde o perito forense faz a sua análise e retira conclusões.
6. Terminada a análise, são carregados os módulos de `report` responsáveis por gerar relatórios finais com as evidencias descobertas pelos módulos de `ingest`.

Conhecidos os principais passos de uma execução do Autopsy durante uma análise forense, analisaremos agora os três diferentes tipos de módulos que integram a ferramenta.

6.1.1 *Datasource processor*

Os módulos do tipo `DataSource Processors` são responsáveis por agregar e preparar as fontes de dados associadas à investigação. São utilizados numa fase primordial do caso.

É possível, com objetivo de satisfazer os requisitos não suportados nativamente pelo Autopsy, programar um módulo deste tipo para ser integrado. Este tipo de módulos está limitado no seu desenvolvimento, suportando apenas programação usando a linguagem `Java` , contrariamente aos outros tipos de módulos (identificados nas secções seguintes) que apresentam também suporte para a linguagem `Python`.

A comunidade *open-source* contribui ativamente para aumentar o modo de como são adicionadas novas fontes de dados. Exemplos de novos módulos de `DataSource Processors` incluem a possibilidade de extracção de conteúdos em tempo real de dispositivos móveis Android[2], extração de *dumps* de memoria RAM[3], entre outros.

A Figura 19 mostra o painel de escolha de alguns módulos de `Datasource processor` no Autopsy.

2 https://github.com/labcif/FAMA
3 https://github.com/volatilityfoundation/volatility

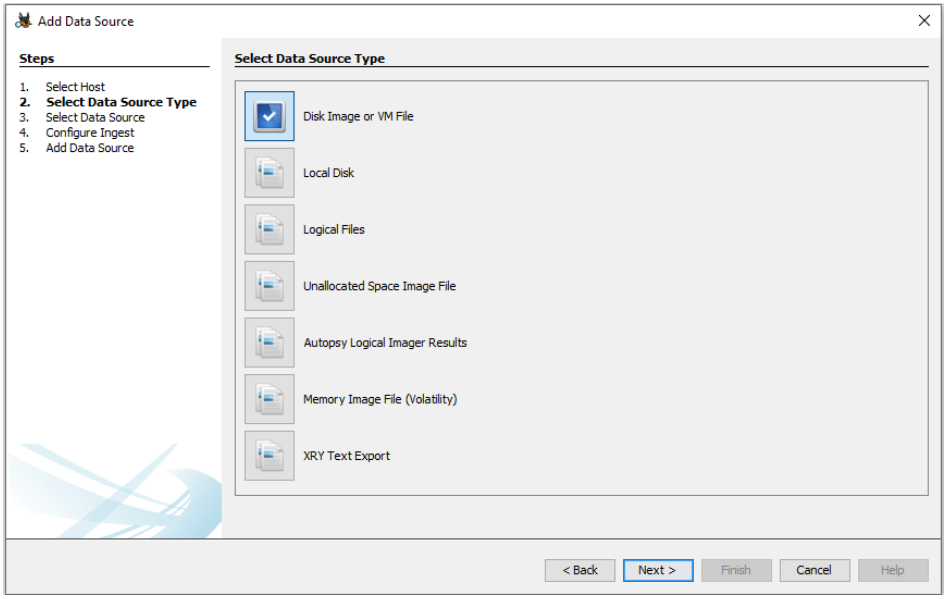

Figura 19: Módulos de Datasource Processor no Autopsy

6.1.2 *Ingest*

Os módulos denominados de `ingest` são responsáveis por realizar toda a análise dos ficheiros previamente carregados como fonte de dados.

No decorrer a execução do módulo, são gerados artefactos forenses e adicionados à base de dados interna do caso. Os artefactos vão sendo adicionados à árvore de artefactos (Figura 20), apresentando resultados para o perito forense.

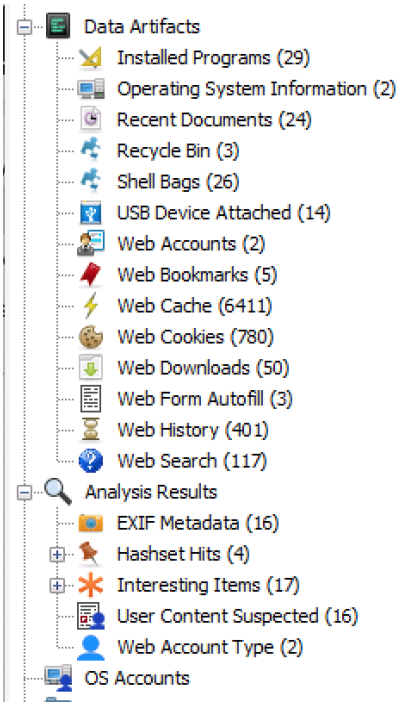

Figura 20: Árvore de artefactos do Autopsy

O programador deste tipo módulos não faz desenvolvimento para interface gráfica do Autopsy, exceto o painel de configurações exibido antes da sua execução.

É este tipo de módulos que a comunidade *open-source* mais desenvolve. Por ser verdade, o Laboratório de Cibersegurança e Informática Forense (LABCIF)[4] do Instituto Politécnico de Leiria (IPL) tem contribuindo ativamente para o desenvolvimento de ferramentas e módulos no âmbito da informática forense. Estes módulos têm sido reconhecidos pela comunidade forense, tendo sido destacados em concursos internacionais de desenvolvimento de módulos como o Open Source Digital Forensics Conference (OSDFCon)[5].

À data da escrita deste documento, a versão usada neste projeto (4.19.1) suporta nativamente 13 módulos de `ingest`, sendo a importação de módulos da comunidade uma prática comum para aumentar as potencialidades do software.

A Figura 21 mostra o painel de configuração dos módulos de `ingest`.

4 https://github.com/labcif
5 https://www.osdfcon.org/

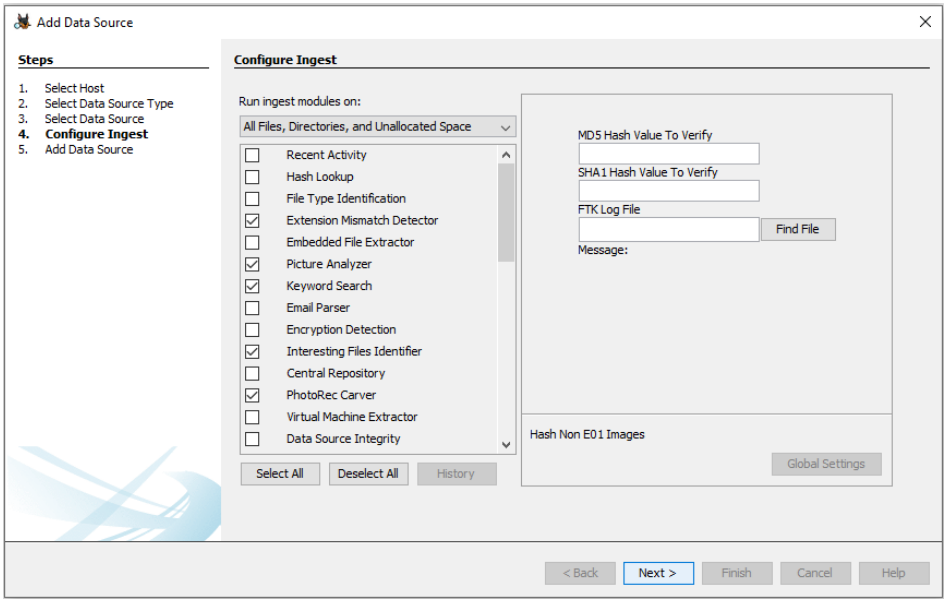

Figura 21: Módulos de Ingest no Autopsy

De notar que podem ser selecionados mais que um módulo de `ingest` em simultâneo. Diferentes módulos identificarão artefactos forenses distintos para a mesma fonte de dados, visto cada módulo ser programado para identificar temáticas diferentes. Para tornar mais célere a apresentação dos artefactos forenses, o Autopsy executa os módulos de `Ingest` em paralelo para todos os ficheiros da fonte de dados.

Foi desenvolvimento um módulo de `ingest` no âmbito deste projeto. O módulo desenvolvido está documentado em detalhe no Capítulo 7.

6.1.3 *Report*

Compilada toda a informação recolhida pelos módulos de `ingest`, seguem-se os módulos de `report`, responsáveis por apresentar os resultados ao perito.

A barreira da linguagem de programação comum aos dois módulos enumerados nas secções anteriores (`datasource processor` e `ingest`) não se aplica no módulo de `report`.

Alguns destes módulos permite gerar um relatório em formato de texto, transformar as coordenadas GPS em ficheiro Keyhole Markup Language (KML), relatórios em formato excel, só para enumerar alguns.

A Figura 22 apresenta o painel de escolha dos módulos de `report`.

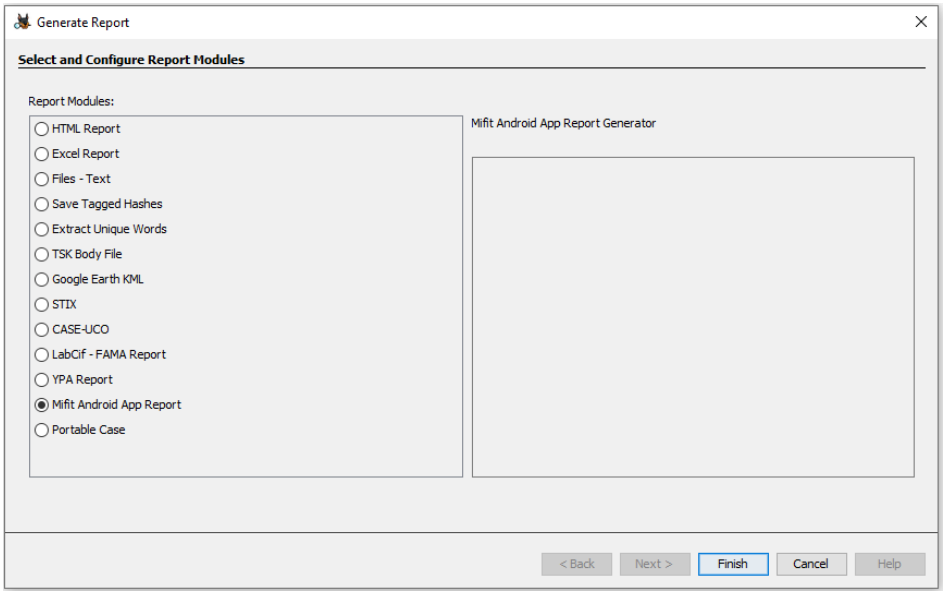

Figura 22: Módulos de Report no Autopsy

O desenvolvimento de módulos de `report` visam cobrir as necessidades de apresentação dos factos encontrados. O perito forense deve escolher o formato que mais se adeque ao tipo de informações recolhidas.

Com objetivo de cobrir todas as necessidades exigidas para este projeto, foi também desenvolvido um módulo de `report`. O módulo desenvolvido está documentado em detalhe no Capítulo 7.

6.2 SUMÁRIO

Este capítulo abordou o software Autopsy, uma ferramenta *open-source* de elevado relevo para a informática forense, destacando-se pela sua extensibilidade e funcionalidades nativas. O Autopsy destaca-se ainda pela sua capacidade de gerar e apresentar artefactos forenses, focalizando-se nas análises *post-mortem*.

O capítulo seguinte realça a plataforma de análise aos conteúdos da aplicação MiFit para Android, desenvolvida no âmbito deste projeto.

7

MIFIT ANALYZER

A análise forense digital a aplicações Android é vulgarmente solicitado durante uma perícia forense. É de destacar a plataforma Forensic Analysis for Mobile Apps (FAMA)[1] integrada no LABCIF desenvolvida no âmbito de um projeto informático na licenciatura de Engenharia Informática. A plataforma FAMA assente em regime *open-source* apresenta-se como uma solução de extração dos artefactos do dispositivo Android e a possibilidade de análise recorrendo à programação de um módulo de análise contendo a lógica especifica a cada aplicação (Francisco et al., 2020).

Com objetivo de acelerar o processo de desenvolvimento e focar os esforços na aplicação MiFit, parte do código desenvolvido na ferramenta FAMA serviu de base de implementação para algumas funcionalidades deste projeto, complementado com melhorias técnicas enquadradas nas necessidades da nova plataforma desenvolvida, o MiFitAnalyzer.

Este capítulo inicia-se com a descrição dos principais objetivos da plataforma MiFitAnalyzer, seguida da arquitetura que a compõe.

7.1 OBJETIVOS

O objetivo principal da ferramenta criada é disponibilizar uma solução centralizada capaz de gerar artefactos forenses inseridos na aplicação MiFit de forma rápida, concisa e fiável. O facto do código estar disponível e usar tecnologias não proprietárias, contribui positivamente para a comunidade *open-source*.

A ferramenta MiFitAnalyzer consegue interpretar o tipo de dispositivos que se conectaram à aplicação MiFit e que informações são armazenadas a quando dessa conexão.

Como referido no Capitulo 5, associada a aplicação está pelo menos uma conta de utilizador. Esta ferramenta objetiva recolher e interpretar o máximo de dados e meta-dados dos utilizadores associados ao dispositivo em análise. Com estes

1 https://github.com/labcif/FAMA

dados, cabe ao perito forense traçar um perfil de atividades do utilizador ou usar a informação recolhida da maneira que considerar mais relevante.

7.2 EXECUÇÃO DA FERRAMENTA

Com objetivo de diminuir a dependência de *frameworks* ou SDK externos, foi tido em consideração o uso exclusivo de módulos nativamente suportados pelo SDK desenvolvimento do Python. À data da escrita deste documento, todos os módulos externos foram importados localmente na diretoria do projeto. Exemplo desta importação é o *script* `sqlparse.py` documentado na Secção 7.4.7. Esta abordagem mitiga incompatibilidades entre versões e módulos que compõe a ferramenta. Outra vantagem manifestada por esta abordagem é a operabilidade da ferramenta sem conexão à Internet.

Para a tirar partido das funcionalidades da ferramenta é necessário que o ambiente tenha possibilidade de executar *scripts* Python. A instalação deste ambiente é similar nos três principais sistemas operativos: sistemas Windows, distribuições Linux e MacOS.

A ferramenta desenvolvida MiFitAnalyzer suporta versões Python 3.x.

À data da escrita desde documento, a ferramenta MiFitAnalyzer pode ser executada no terminal do sistema operativo ou como módulo Python externo no Autopsy.

A Listagem 10 mostra o comando para executar a plataforma MiFitAnalyzer no terminal do sistema operativo. De notar que o exemplo abaixo apresentado pressupõe a associação prévia da variável `python` nas variáveis de ambiente do sistema operativo.

```
1    python start.py -p <path/to/dump> -o <report.json>
```

Listagem 10: Comando para executar MiFitAnalyzer no terminal

A Tabela 25 resume os argumentos para executar a plataforma MiFitAnalyzer no terminal do sistema operativo.

ARGUMENTO	ARGUMENTO EXTENSO	DESCRIÇÃO
-p	–path	diretoria a ser analisada
-o	–output	diretoria do relatório final
-h	–help	opções disponíveis / ajuda
-s	–start	data de inicio dos artefactos forenses
-e	–end	data de fim dos artefactos forenses
-g	–gps	gerar ficheiro KML com as coordenadas GPS

Tabela 25: Argumentos para executar a plataforma MifitAnalyser

O parâmetro obrigatório `--path` recebe o caminho relativo ou absoluto da diretoria com os conteúdos da aplicação a partir da diretoria da máquina local onde está a ser executada a aplicação MiFitAnalyzer.

O parâmetro `--output` recebe o caminho relativo ou absoluto para o ficheiro de formato JSON que irá ser gerado com todas as descobertas forenses e artefactos descobertos pela plataforma MiFitAnalyzer. A escolha deste formato de saída de dados agnóstico facilita a leitura das evidencias por outros softwares, independentemente da linguagem de programação usada por estes.

Os parâmetros `--start` e `--end` definem um limite temporais inferior e superior respetivamente. Os artefactos gerados estarão contidos neste intervalo. Para ambos os parâmetros deve ser fornecido uma data no formato dd-mm-aaaa.

A Listagem 11 exemplifica o comando para a execução da plataforma fazendo uso destes parâmetros.

```
1    python start.py --start 01-01-2022 --end 31-12-2022 --path <path/to/dump>
     ↪  -output <report.json>
```

Listagem 11: Comando para executar MiFitAnalyzer no terminal

Para o exemplo mencionado acima, apenas serão indexados artefactos forenses cuja a data que o representa esteja contida entre o dia 01-01-2022 e o dia 31-12-2022. Sendo estes parâmetros opcionais, os valores por omissão definidos para o limite inferior é `01-01-1970`. O instante superior é marcado pelo início da execução da ferramenta MiFitAnalyzer em caso de omissão do parâmetro `--end`. De notar que os parâmetros são tratados de forma isolada, isto é, não necessitam de ser executados em simultâneo.

Como referenciado no Capitulo 5, alguns artefactos forenses caracterizam-se por conter coordenadas GPS. Com objetivo de optimizar uma possível análise que não necessite deste tipo de artefactos, o parâmetro `--gps` recebe um boleano (`True` ou

`False`) desenhado para ativar ou desativar a associação de coordenadas GPS aos artefactos forenses. Uma vez que o perito forneça o parametro a `True`, será gerado um ficheiro em formato KML contendo todos os pontos identificados durante a análise. Este parâmetro apresenta o valor `False` por omissão.

O parâmetro `--help` lista no terminal todas as opções possíveis de serem executadas na plataforma MiFitAnalyzer.

7.3 INTEGRAÇÃO COM O AUTOPSY

Foi uma preocupação desde o início do projeto integrar totalmente a ferramenta desenvolvida com o software de análise forense Autopsy. As inúmeras funcionalidades disponibilizadas pelo Autopsy aliadas à lógica de análise forense especifica para a aplicação Android MiFit proporcionada pela ferramenta MiFitAnalyzer, disponibiliza ao perito forense um vasto conjunto de evidencias e artefactos num interface familiar que é a do Autopsy.

Os passos a seguir apresentados explicam como pode ser importada a ferramenta MiFitAnalyzer recorrendo ao software Autopsy para as versões 4.X:

1. Abrir o Autopsy -> Tools -> Python Plugins
2. Extrair os conteúdos da ferramenta para a diretoria `python_modules`
3. Reiniciar o Autopsy

Seguidos os três passos mencionados acima, na próxima execução do Autopsy, a ferramenta MiFitAnalyzer estará totalmente integrada no software, onde se inclui a componente de análise (módulo de Ingest) e a componente para gerar o relatório final dinâmico (módulo de Report).

7.4 ARQUITETURA

As secções seguintes documentam a arquitetura geral da solução proposta. Foram evitadas *frameworks* complexas que requeiram ambientes de execução próprios ou instalação de software externo.

A fluxograma da Figura 23 demonstra os principais componentes da plataforma MiFitAnalyzer e as suas interações.

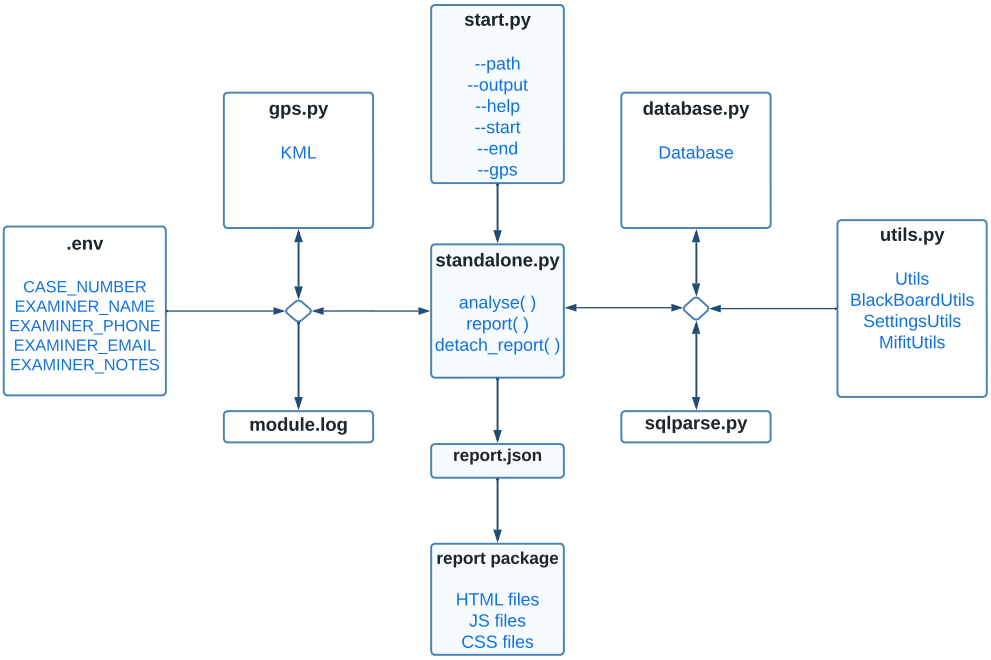

Figura 23: Fluxograma de componentes da plataforma MiFitAnalyzer

As secções seguintes detalham cada um dos componentes subjacentes à plataforma MiFitAnalyzer.

7.4.1 *start*

O *script* `start` é ponto inicial da execução pela linha de comandos responsável pela gestão de parâmetros de entrada para a execução da ferramenta.

7.4.2 *mifit*

O *script* `mifit` é o ponto inicial da execução da ferramenta pelo software de análise forense Autopsy. Nele constam duas classes: `MifitIngestModuleFactory` e `MifitReportModule`. A primeira é responsável por indicar ao Autopsy que estamos perante um módulo de Ingest no momento de carregamento dos módulos deste tipo. Esta classe contém informações acerca do módulo criado, onde se inclui o nome, breve descrição, versão atual e a referência para a classe Python responsável pela indexação dos artefactos forenses gerados no período de análise.

Semelhante à anterior, a classe `MifitReportModule` segue o *template* fornecido na documentação oficial[2] para que este seja reconhecido como um módulo de Report válido e posteriormente seja adicionado à lista dos já existentes no Autopsy.

7.4.3 *.env*

O ficheiro `.env` contém as variáveis de ambiente que permitem identificar o utilizador da plataforma MiFitAnalyzer numa execução pelo terminal. À data da escrita deste documento, são suportadas quatro variáveis (Tabela 26).

VARIAVEL	DESCRIÇÃO
CASE_NUMBER	número identificador do caso
EXAMINER_NAME	nome do perito forense
EXAMINER_PHONE	contacto telefónico do perito forense
EXAMINER_EMAIL	e-mail do perito forense
EXAMINER_NOTES	observações do perito forense

Tabela 26: Variáveis de ambiente da plataforma MiFitAnalyzer

A Listagem 12 exemplifica um ficheiro de configuração `.env` das variáveis de ambiente. O ficheiro de configuração deve ser preenchido previamente (antes do inicio da análise).

Listagem 12: Ficheiro de configuração .env do MiFitAnalyzer

```
1  CASE_NUMBER=1
2  EXAMINER_NAME=Jose Francisco
3  EXAMINER_PHONE=+351910000000
4  EXAMINER_EMAIL=jose.francisco@examiner.com
5  EXAMINER_NOTES=Forensic analysis of mifit app
```

No caso de uma execução iniciada pelo Autopsy, a plataforma MiFitAnalyzer importa automaticamente os valores introduzidos na interface gráfica do software, mapeando as configurações sem necessidade de serem introduzidas novamente no ficheiro `.env`.

A Figura 24 mostra o painel de configuração dos dados do perito forense e do caso para a versão 4.19 do Autopsy.

2 http://sleuthkit.org/autopsy/docs/api-docs/4.19.3//mod_python_report_tutorial_page.html

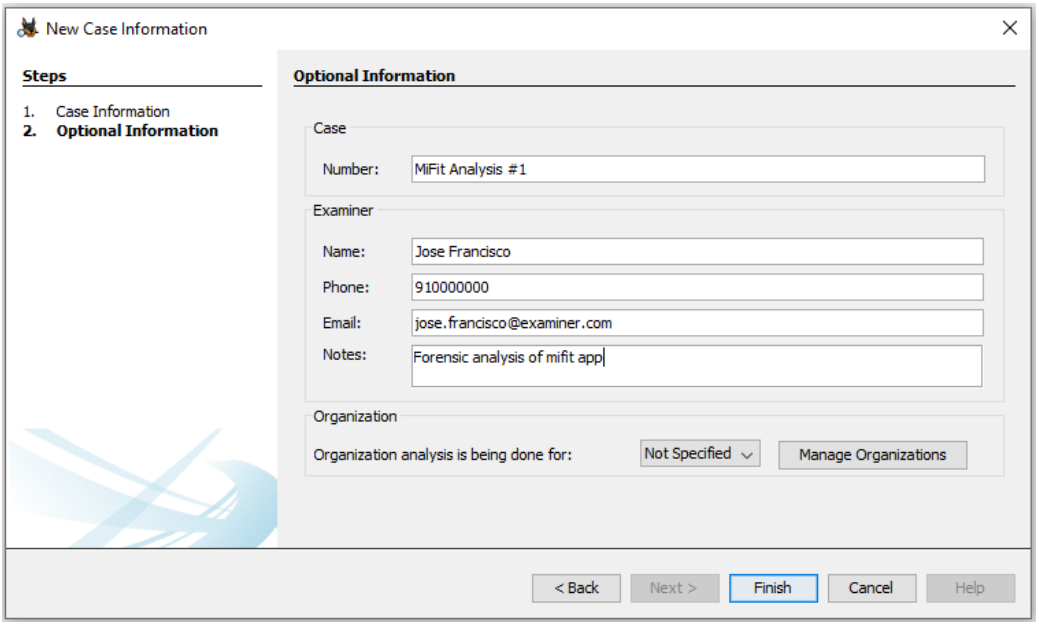

Figura 24: Painel de configuração do caso no Autopsy

7.4.4 *standalone*

O *script* `standalone` tem a responsabilidade de processar os ficheiros da aplicação MiFit previamente carregados pelo Autopsy ou fornecidos no parâmetro `--path` no caso da execução pelo terminal.

A classe `Standalone` integrada no *script* engloba diversos métodos públicos e privados com lógica de análise dos ficheiros identificados como fonte de artefactos forenses. Entre os métodos publicos, destacam-se três, identificados na Tabela 27.

MÉTODO	CARACTERÍSTICA	DESCRIÇÃO
analyse()	método da classe	iniciar a análise dos conteúdos da aplicação
report()	método da classe	gerar relatório dinâmico
detach_report(report, report_path)	estático	gerar relatório dinâmico baseado num relatório JSON

Tabela 27: Métodos públicos da classe Standalone

Após o carregamento dos conteúdos é criada uma instância da classe `Standalone` recorrendo ao seu construtor.

A análise forense inicia-se após a chamada do método `analyse()`. Este método recolhe os ficheiros relevantes, auxiliando-se dos métodos privados da classe

`Standlalone` para os analisar. Nestes ficheiros para análise incluem-se bases de dados, *shared_preferences* entre outros. É também da responsabilidade do método `analyse` carregar as variáveis do caso definidas no ficheiro `.env` ou os dados do caso introduzidas no Autopsy.

O método `report()` gera os conteúdos necessários para a apresentação do relatório dinâmico em HyperText Markup Language (HTML). Do ponto vista técnico, este método gera uma variável `JavaScript` e grava-a no ficheiro `report.js`. Posteriormente, o ficheiro `report.js` é consumido pelo módulo de report, que contém toda a lógica de apresentação dos artefactos forenses encontrados durante todo o processo de análise.

O resultado da função estática `detach_report(report,report_path)` é análogo ao método `report()` descrito anteriormente. A particularidade deste método é refletida na sua chamada. Sendo uma função estática, não é necessária uma instância da classe `Standalone` para ser executada. A função recebe dois argumentos: *i*) report: conteúdo de um relatório em formato JSON gerado por uma análise, *ii*) report_path caminho para a diretoria onde irá constar os conteúdos do relatório dinâmico em HTML.

7.4.5 *settings*

O *script* `settings` contém duas classes: `MifitIngestSettingsPanel` e `MifitReportSettingsPanel` responsáveis por renderizar os painéis de configuração dos módulos de Ingest e Report respetivamente.

A classe `MifitReportSettingsPanel` não requereu qualquer implementação para além da declaração da classe.

Sendo o módulo maioritariamente desenvolvido na linguagem Python, o painel de configuração do módulo de Ingest servido pela classe `MifitIngestSettingsPanel` é desenvolvido com recurso à tecnologia `Jython`.

A Figura 25 reflete o painel de configurações do módulo de Ingest da plataforma `MiFitAnalyser` no software Autopsy.

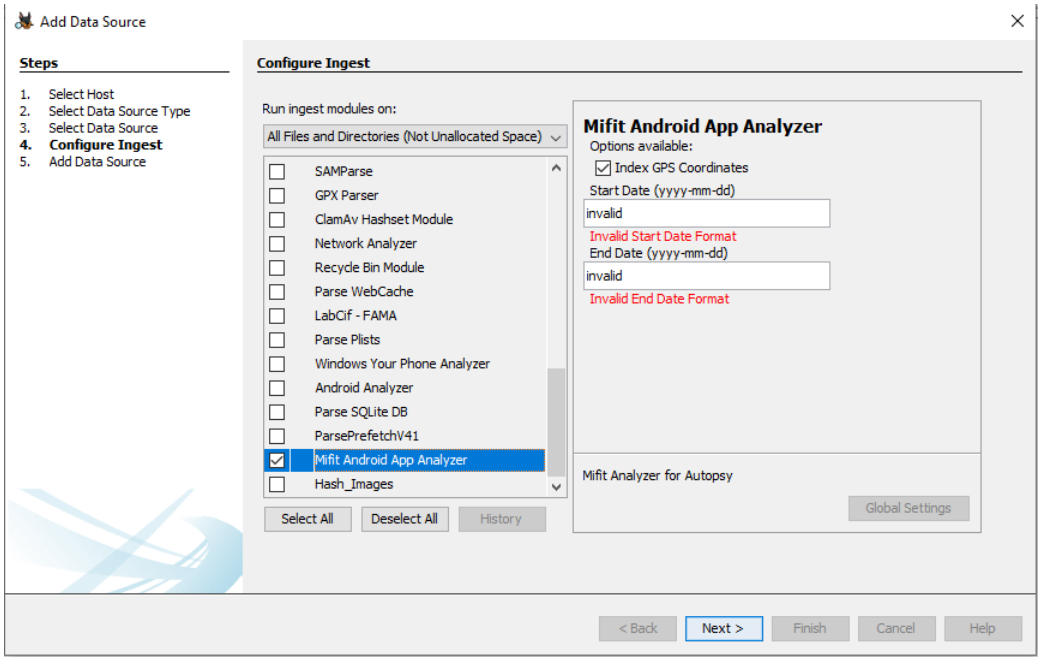

Figura 25: Painel de configuração do módulo MiFitAnalyzer

A MiFit captura coordenadas GPS para apresentar algumas das suas funcionalidades. Dada a possibilidade de existir uma grande quantidade de artefactos forenses deste tipo passíveis de serem indexados, o tempo de análise naturalmente será maior. De modo a tornar as tarefas o mais céleres e eficientes possível, o perito forense pode optar por não indexar artefactos provenientes de coordenadas geográficas caso estas existam na fonte de dados fornecida. De notar que os artefactos serão visíveis na árvore de artefactos do *software* Autopsy, apenas não serão associadas ao mapa nativamente disponível no Autopsy.

O perito forense pode ainda indicar um intervalo temporal, indexando apenas artefactos forenses que estejam contidos nesse período. O perito deve indicar uma data inicial e uma data final em formato yyyy-mm-dd nas caixas de texto correspondentes. Caso as datas apresentem um formato inválido, será despoletado um aviso como mostra a Figura 26, ignorando as datas inseridas.

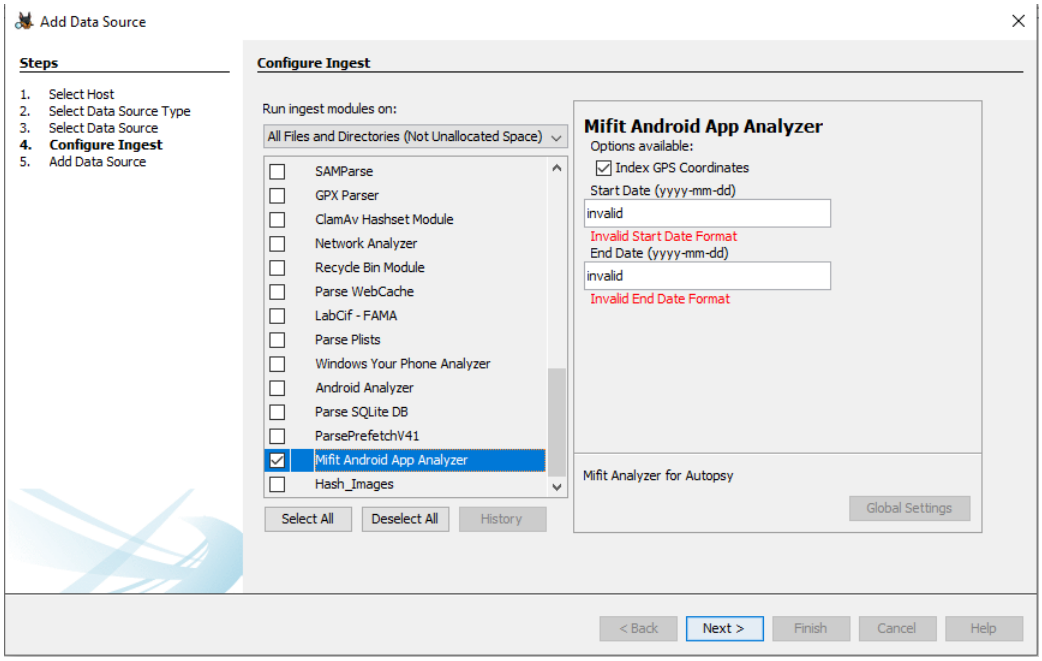

Figura 26: Mensagens de erro no módulo MiFitAnalyzer

7.4.6 *ingest*

O *script* `ingest` é o principal responsável pela indexação dos artefactos forenses no software Autopsy. Seguindo os padrões recomendados para o desenvolvimento de módulos Ingest, o *script* inclui a classe `MifitIngestModule`. A Tabela 28 descreve sucintamente os métodos da classe `MifitIngestModule`.

MÉTODO	DESCRIÇÃO
___init___()	construtor da classe
get_info(attr)	obter dinamicamente parte do relatório
startUp(report, report_path)	pré-análise
process(dataSource, progressBar)	análise
shutdown()	pós-análise

Tabela 28: Métodos públicos da classe MifitIngestModule

Uma vez que o Autopsy não suporta o tipo de artefactos forenses passíveis de serem extraídos da aplicação MiFit, compete ao *script* ingest cria-los.

A Tabela 29 descreve os artefactos criados no Autopsy.

IDENTIFICADOR	ARTEFACTO	DESCRIÇÃO
heartRate	MIFIT Heart Rate Records	medição do batimento cardíaco
alarm	MIFIT Alarms	alarme programado
steps	MIFIT Steps	registo de passos
sleep	MIFIT Sleep	registo do sono
workout	MIFIT Workouts	atividade desportiva
userInfo	MIFIT User Info	informações do utilizador
stress	MIFIT Stress	registo de stress
devices	MIFIT Devices	dispositivos associados

Tabela 29: Artefactos criados pelo MiFitAnalyzer no Autopsy

Criados os tipos de artefactos, estes têm de ser populados com dados recorrendo a atributos. Entende-se por atributos as características ou propriedades dos artefactos. De notar que os atributos podem ser reutilizados em diferentes tipos de artefactos.

A Tabela 30 descreve os atributos criados no Autopsy.

IDENTIFICADOR	ATRIBUTO	DESCRIÇÃO
heartRate	Heart Rate	valor numérico do batimento cardíaco
enabled	Enabled	0=inativo, 1=ativo
start	Start	início
stop	Stop	data de fim
mode	Mode	modo
distance	Distance	distância
calories	Calories	calorias
steps	Steps	número de passos
type	Type	tipo
datestr	Date	data em formato string
cadence	Cadence	cadência
startTime	Start Time	data de início
endTime	End Time	data de fim
from	From	de (remetente)
to	To	para (destinatário)
provider	Provider	fornecedor
registDate	Regist Date	data de registo
appToken	App Token	token de aplicação
loginToken	Login Token	Token de autenticaçãoo
idToken	Token Id	identificador de um token
thirdId	Third Party Id	identificador de fornecedores externos
stress	Stress Value	valor de stress

Tabela 30: Atributos criados pelo MiFitAnalyzer no Autopsy

Criados os tipos de atributos, são lidas as configurações enviadas previamente pelo *script* `settings`, criando uma nova instância da classe `standalone` responsável pela análise. No decorrer da análise, a instância criada gera o relatório em formato JSON.

O método `process(dataSource,progressBar)` processa a análise feita, mapeando as informações recolhidas para os artefactos e atributos criados para o Autopsy.

O método `shutDown()` liberta recursos alocados durante o processo de análise.

7.4.7 *database*

Como verificado no Capítulo 5.2, as bases de dados constituem uma importante fonte de dados no âmbito de uma análise forense a dispositivos Android. Sendo estes dados significativos, é fundamental criar mecanismos para extracção dos mesmos de forma rápida e fiável.

Em virtude do projeto ser fundamentalmente implementado com recurso à linguagem `Python`, contrastando com a linguagem `Java` presente no software Autopsy, sentiu-se a necessidade de criar uma interface única que garantisse o suporte em ambas as tecnologias.

O *script* `database.py` interage diretamente com os ficheiros de base de dados SQLite3, permitindo que sejam executadas operações de Data Manipulation Language (DML) sobre elas usando a linguagem Structured Query Language (SQL). Dependendo do ambiente onde é executada a plataforma MiFitAnalyzer, os conectores às bases de dados diferem. Para um ambientes de execução baseados na linguagem `Python`, onde se inclue a versão que pode ser executada pelo terminal do sistema operativo, é usado o *package* sqlite3[3] nativamente suportado pelo Python. Para ambientes que recorrem à linguagem Jython, onde se inclui o *software* de análise forense Autopsy, é usado o conector `jdbc:sqlite`[4].

De notar que o Autopsy injeta o conector `jdbc:sqlite` no *script* `database.py`, sendo apenas necessário adicionar lógica para importar os conectores dinamicamente baseado na plataforma de execução.

A classe `Database` engloba ainda métodos que permitem a recuperar registos apagados ou corrompidos nas bases de dados recorrendo a duas ferramentas *open-*

3 https://docs.python.org/3/library/sqlite3.html
4 https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc

source: `undark`[5] e `SQLite Deleted Records Parser`[6]. A ferramenta `undark` baseada na linguagem `C`, gera uma saída em formato Comma Separated Values (CSV) com os registos parciais obtidos. A ferramenta `SQLite Deleted Records Parser` complementa a ferramenta supracitada com a análise espaço não alocado e blocos livres da base de dados.

7.4.8 *gps*

As coordenadas geográficas (GPS) revelaram-se uma temática importante neste projeto. O *script* `gps.py` gere e facilita a captura, registo e apresentação das coordenadas GPS encontradas durante a análise forense aos conteúdos da aplicação MiFit. Contendo uma única classe: `Kml`, representativa de uma sequência de coordenadas em formato KML. A escolha deste formato justifica-se por ser suportado por inúmeras plataformas de mapas, incluindo as gigantes `Google Maps`[7], `Bing Maps`[8], `Open Street Maps`[9], só para enumerar algumas, tendo sido aprovado em 2008 pelo Open Geospatial Consortium[10] como uma norma internacional para representar coordenadas geográficas (Sandvik, 2008).

7.4.9 *utils*

No ficheiro `utils` são persistidos blocos de código e métodos, podendo estes ser importados por outras classes em diferentes *scripts* do projeto. O ficheiro `utils` é composto por quatro classes: `Utils`, `BlackBoardUtils`, `SettingsUtils` e `MifitUtils`. A escolha da separação em quatro classes distintas deve-se ao facto de os métodos nelas contidas apresentarem comportamentos e grupos de funcionalidades dstintas, tornando a sua importação mais eficiente e facilitando o programador na implementação de módulos a desenvolver no futuro.

Os métodos têm uma implementação estática com nomenclatura adequada à sua funcionalidade facilitando assim a legibilidade, removendo a necessidade de criar uma instância da classe.

Os parágrafos seguintes descrevem as quatro classes presentes no ficheiro `utils`.

5 http://pldaniels.com/undark
6 https://github.com/mdegrazia/SQLite-Deleted-Records-Parser
7 https://www.google.com/maps
8 https://www.bing.com/maps
9 https://www.openstreetmap.org
10 https://www.ogc.org/pressroom/pressreleases/857

utils. A classe `Utils` é formada por 19 métodos de auxílio geral, onde se incluem leitura e escrita de ficheiros, transformação de formatos de data, inicialização de sistemas de logs, entre outras funcionalidades. A Tabela 31 apresenta os métodos associado a uma descrição sumária da sua funcionalidade.

MÉTODO	Parâmetros	DESCRIÇÃO
setup_custom_logger	logfile	configuração do sistema de logs centralizado
setup_case	-	ler metadados do caso em curso
list_files	source, pattern, exclude	listar ficheiros que correspondem aos parâmetros de entrada
get_base_path_folder	-	ler diretoria do ficheiro atual
check_and_generate_folder	path	criar uma diretoria caso não exista
remove_folder	folder	remover direorias e subdiretorias de forma segura
read_json	path	leitura de ficheiro em formato JSON
write_json	path, contents	escrita em ficheiro em formato JSON
verify_header_signature	file, header_type, offset, stream	verificar tipo do ficheiro baseado na sua assinatura
minutes_to_time	minutes	conversão de minutos para formato hh:mm
timestamp_to_time	timestamp	conversão de Unix Timestamp para formato aaaa:mm:ss hh:mm:ss
xml_attribute_finder	xml_path, attrib_values	obter valores contidos em ficheiros Extensible Markup Language (XML) de forma iterativa
get_autopsy_version	-	obter versão do autopsy
copy_tree	src, dst	copiar diretorias, subdiretorias e ficheiros de forma iterativa
date_to_timestamp	date, date_format, default	conversor de data para Unix Timestamp
is_timestamp_between_timestamps	timestamp, lower_limit, upper_limit	verificar se Unix Timestamp está contido noutros dois
get_current_timestamp	-	obter Unix Timestamp atual
get_current_date	date_format	obter data atual dado formato
is_valid_date	date, date_format	verificar se uma data é valida para o formato dado

Tabela 31: Métodos auxiliares da classe Utils

blackboardutils. Na classe `BlackBoardUtils` residem métodos auxiliares associados ao `Blackboard` do Autopsy.

Com o objetivo de perceber a necessidade da criar esta classe auxiliar que interage com o *BlackBoard* do Autopsy, os parágrafos seguintes apresentam uma explicação sumária do seu funcionamento.

Artifacts. Tipicamente um artefacto forense é interpretado como o conteúdo parcial ou total de dados que apresentem relevância do ponto de vista forense.

Fragmentos de uma imagem, documentos, mensagens, registos de *logs* são alguns exemplos desses artefactos. No software Autopsy, a representação desta informação é apresentada na forma de categorias/tipos de artefactos. Os artefactos com características semelhantes são agrupadas, e representados na árvore de artefactos presente na lateral esquerda do ecrã principal (Figura 27). Os atributos inerentes a cada tipo de artefacto são representados na forma de colunas contendo dados e meta-dados, estruturando assim o artefacto forense.

A título de exemplo, foi criado o tipo de artefacto forense representativo das atividades desportivas do utilizador intitulado de `MIFIT - Atividades Desportivas`, onde engloba os atributos: `utilizador`, `dispositivo`, `início`, `fim`, `coordenadas`, só para enumerar alguns. Cada linha da tabela do painel principal representa uma atividade desportiva praticada pelo utilizador.

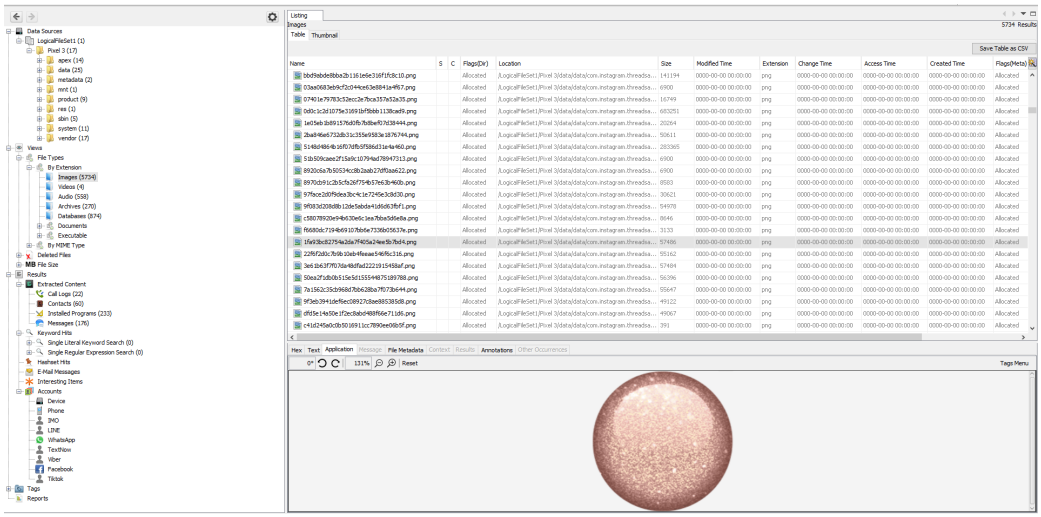

Figura 27: Painel principal de artefactos do software Autopsy

Communication. No Autopsy, as comunicações podem ser estabelecidas por diferentes vias, onde se incluem mensagens ditas curtas, como as SMS, troca de emails ou mensagens diretas em redes sociais por exemplo. O separador de comunicações apresenta um grafo onde cada vértice representa os intervenientes da comunicação e os segmentos de reta as suas ligações ponto-a-ponto (Figura 28). É ainda possivel filtrar as contas mais usadas ou apresentar apenas comunicações dentro de um período de temporal específico.

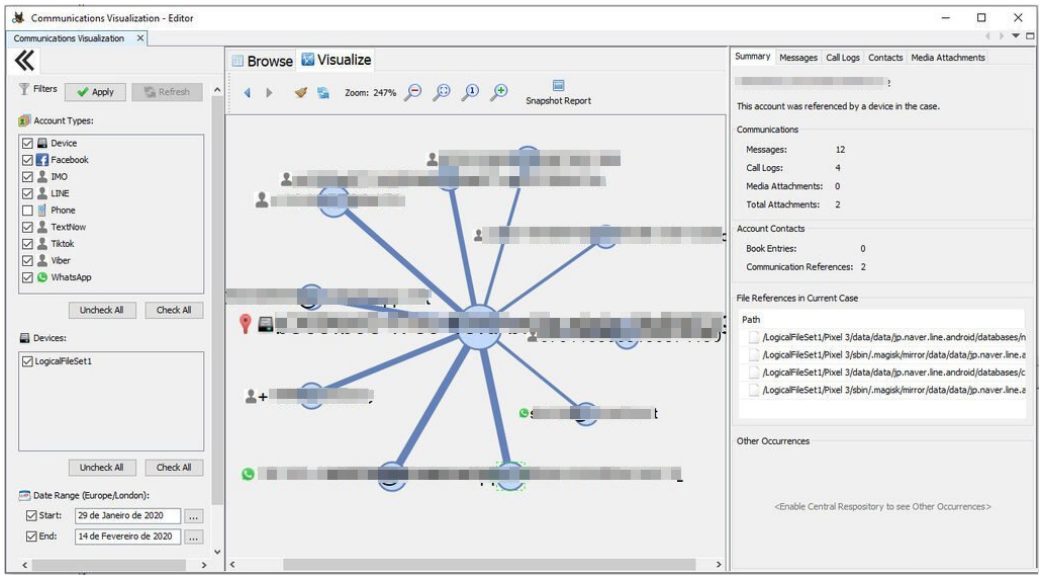

Figura 28: Separador de Comunicações do Autopsy

Geolocation. Para fazer face à necessidade de representar graficamente os artefactos forenses gerados recorrendo a coordenadas GPS, o separador `geolocation` apresenta um mapa dinâmico. Ao adicionar os pontos de interesse, estes ficam registados no mapa, sendo possível filtrar-los por fonte de dados ou por instante temporal que está associado às coordenadas do artefacto.

Como exemplo, Figura 29 mostra os marcadores azuis representativos das coordenadas GPS adicionadas. Por omissão, na lateral esquerda do separador `geolocation` é apresentado o menu de contexto para filtragem dos objetos gerados.

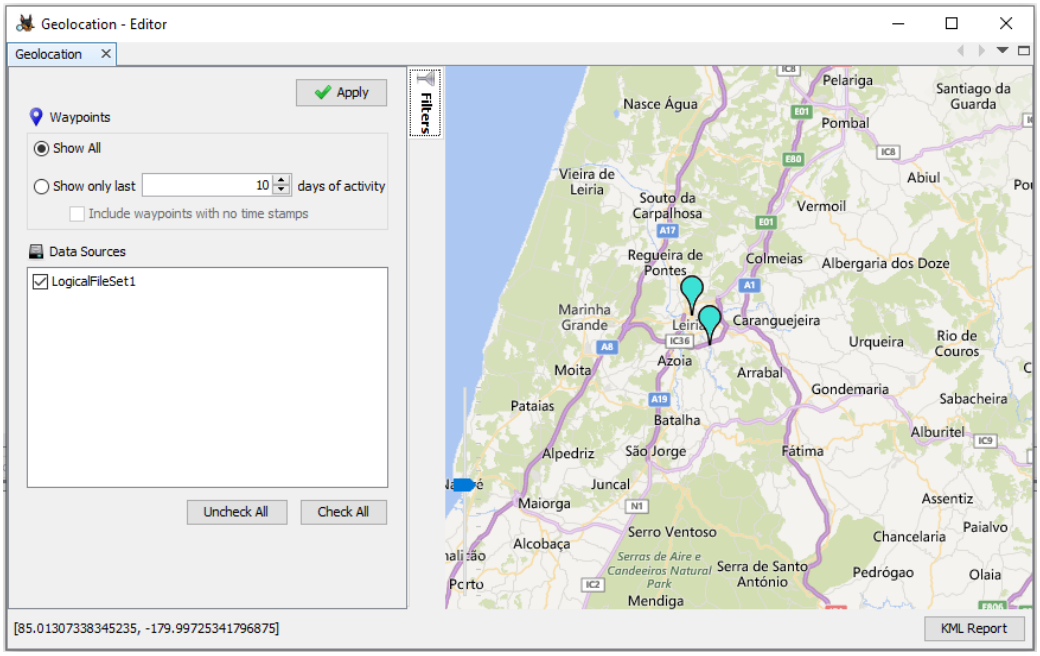

Figura 29: Separador de Geolocalização do Autopsy

Timeline. A representação dos artefactos organizada de forma cronológica poderá revelar-se interessante para compreender a sequência normal dos eventos ocorridos. Essa sequência de artefactos organizada temporalmente é denominada de `timeline` no contexto do software Autopsy. Esta funcionalidade consiste no armazenamento dos artefactos numa lista ordenada por um instante temporal. A Figura 30 mostra o separador `Timeline` no software Autopsy.

Semelhante a outras funcionalidades, é possível aplicar filtros na informação apresentada, cingindo apenas ao conteúdo relevante,

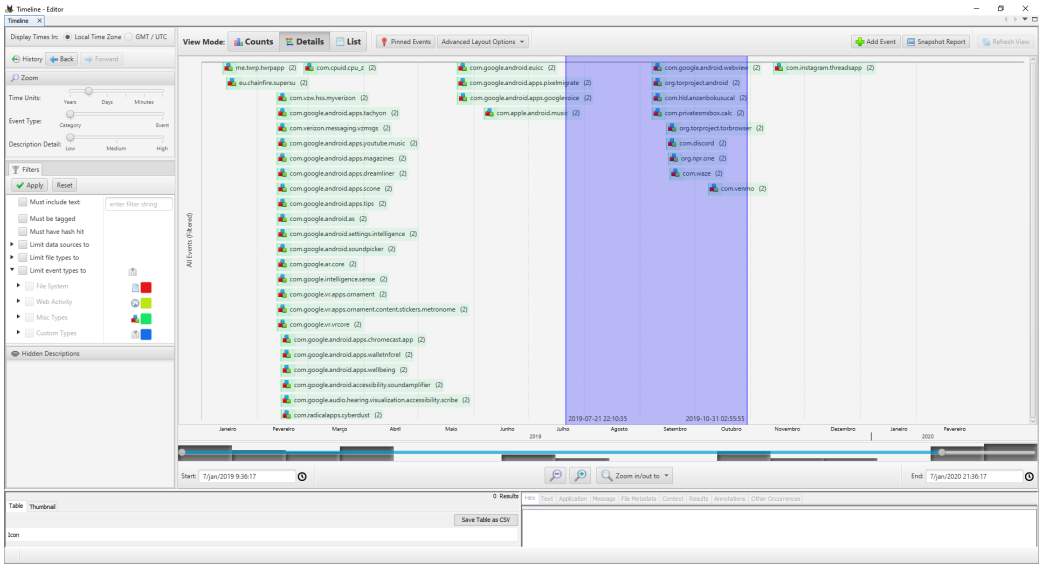

Figura 30: Separador de linha temporal do Autopsy

Clarificados alguns dos principais componentes do software Autopsy, a Tabela 32 apresenta a lista de métodos criados na classe `BlackBoardUtils`.

MÉTODO	Parâmetros	DESCRIÇÃO
post_messager	msg	alterar mensagem do estado global
create_attribute_type	att_name, att_desc	criar novo tipo de atributo
create_artifact_type	base_name, art_name, art_desc	criar novo tipo de artefacto
get_artifacts_list	-	obter lista de artefactos indexados
index_artifact	artifact, artifact_type, attributes	indexar artefacto no Blackboard
add_relationship	node1, node2, art, relationship_type, timestamp	estabelecer ligação entre artefactos (ex: *Communication)*
add_tracking_point	file, timestamp, latitude, longitude, altitude, source	Adicionar coordeanda GPS ao mapa
get_or_create_account	account_type, file, uniqueid	criar artefacto forense do tipo `Account`
add_account_type	accountTypeName, displayName	criar novo tipo de conta para o artefacto forense `Account`

Tabela 32: Métodos auxiliares da classe BlackBoardUtils

Esta classe funciona como uma interface facilitadora para interagir com os componentes graficos nativos do Autopsy, já que este suporta um vasto conjunto de tipos de artefactos forenses. Recorrendo a estes métodos, é possível estabelecer relações entre artefactos do mesmo tipo. Exemplos destas funcionalidades de relação são os importantes artefactos forenses denominados de `Geolocation`, `Timeline`, `Accounts`, só para enumerar alguns.

settingsutils. A classe `SettingsUtils` integra os métodos auxiliares para desenhar a interface gráfica do painel de configurações do módulo de Ingest.

Os métodos contidos na classe, apesar de serem codificados na linguagem Python, são totalmente passiveis de serem executados em ambientes Jython, uma implementação Java Virtual Machine (JVM) da linguagem de programação Python, como requerido pelo Autopsy.

A principal vantagem de usar a tecnologia `Jython` é a capacidade de desenhar uma interface gráfica projetada em Python, usando elementos de Abstract Window Toolkit (AWT)[11] e Swing[12] da linguagem `Java` no Autopsy.

Os métodos da classe `SettingsUtils` podem ser consultados na Tabela 33.

MÉTODO	Parâmetros	DESCRIÇÃO
createPanel	scroll, ptop, pleft, pbottom, pright	criar novo painel
createRadioButton	name, ac, ap	criar campo do tipo `RadioButton`
createInfoLabel	text	criar etiqueta informativa
createSeparators	count	criar espaços / separadores
createCheckbox	label, ap, visible	criar campo do tipo de seleção
createInputField	ap, enabled	criar campo de texto

Tabela 33: Métodos auxiliares da classe SettingsUtils

mifitutils. A classe `MifitUtils` está associada a lógica de negócio da aplicação MiFit. Semelhante às classes auxiliares mencionadas anteriormente, a classe contém métodos auxialiares, no entanto, estes apenas farão sentido no contexto da aplicação MiFit. A Tabela 34 descreve os cinco métodos presentes na classe `MifitUtils`.

MÉTODO	Parâmetros	DESCRIÇÃO
mode_to_workout	mode	Mapeador do tipo de atividade desportiva para o nome da atividade desportiva
mode_to_activity	mode	Mapeador de modo de atividade para o nome da atividade (Tabela 18)
mode_to_sleep	mode	Mapeador de modo de sono para o tipo de ciclo de sono (Tabela 15)
getCoordianateByString	raw	transformação de coordenadas textuais em coordenadas numéricas
getCoordinateByBulkArray	bulk	transformação de conjunto de coordenadas em coordenadas de texto

Tabela 34: Métodos auxiliares da classe MifitUtils

11 https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html
12 https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html

7.4.10 *report*

A diretoria `report` contém todos os recursos necessários à visualização do módulo de report desenvolvido no âmbito da ferramenta MiFitAnalyzer.

O relatório final dinâmico são um conjunto de ficheiros na linguagem de programação JavaScript e a linguagem de marcação HTML amplamente usadas atualmente em páginas web. Não foram utilizadas *frameworks* complexas que requeiram um ambientes de execução próprios.

Para a visualização dos conteúdos gerados módulo de Report, o perito forense poderá recorrer a um browser convencional (exemplo: Google Chrome[13], Mozilla Firefox[14], Microsoft Edge[15]).

A Tabela 35 lista os ficheiros usados pelos módulo de Report associados à sua responsabilidade dentro do mesmo.

FICHEIRO HTML	FICHEIRO JS	DESCRIÇÃO
alarms.html	alarms.js	alarmes configurados pelo utilizador
heart-rate.html	heart-rate.js	histórico dos registos do batimento cardíaco
-	navbar.js	barra de navegação do módulo de report
index.html	profiles.js	informações do utilizador
sleep.html	sleep.js	tabela com fases do sono do utilizador
steps.html	steps.js	gráfico representativo dos passos do utilizador
stress.html	stress.js	gráfico representativo de stress do utilizador
spo2.html	spo2.js	histórico dos registos da saturação de oxigénio (spo2)
workouts.html	workouts.js	tabela de atividades do utilizador, incluindo mapa com coordenadas GPS

Tabela 35: Recursos para o módulo de report

Foi uma preocupação tornar a análise do relatório final o mais célere e eficaz possível. De modo a atingir esse objetivo, foram tomadas decisões técnicas para aumentar o desempenho na análise, exemplificando:

- Todas as tabelas que contenham dados encontram-se paginadas de modo a facilitar a legibilidade.
- Pesquisa e filtragem em todas as colunas que compõem a tabela
- Possibilidade de ordenar valores em todas as colunas

13 https://www.google.com/chrome/
14 https://www.mozilla.org/en-US/firefox/new/
15 https://www.microsoft.com/en-us/edge

A barra de navegação principal representada à esquerda lista as ligações para os separadores do módulo de Report. Cada separador contém informação específica para cada tipo de artefacto forense.

A quantidade de artefactos gerados poderá dificultar a análise do relatório dinâmico. Se por um lado a quantidade for reduzida, a falta de informação poderá não trazer informações benéficas para a investigação. Em contraste, o excesso de artefactos poderá abstrair o perito forense do essencial, contendo informação desnecessária e irrelevante. Foi a pensar nesta situação que existe a possibilidade de limitar inferior e superiormente as datas associadas aos artefactos forenses, ainda que a análise dos conteúdos da aplicação tenha sido feita usando outros intervalos temporais.

As datas são guardadas na `localStorage`[16] do browser. Com esta abordagem, o perito forense pode recarregar a página, navegar entre separadores ou sair do browser sem que a informação seja perdida, garantindo assim que as configurações das datas definidas anteriormente estarão salvaguardadas. Caso o utilizador queira reiniciar a configuração das datas, o botão Clear Dates serve esse propósito, colocando a data inicial a 01-01-2014 e a data final o instante atual (Figura 31). Deste modo, serão apresentados todos os artefactos capturados durante o processo de análise.

Figura 31: Seletor de datas no MiFitAnalyzer

As informações subjacentes ao caso, onde se inclui os dados do perito forense, número do caso e respetivas notas são visíveis canto inferior esquerdo do relatório dinâmico, como demonstrado na Figura 32.

16 Zona de armazenamento local no browser que permite guardar objetos usando uma tipologia chave-valor.

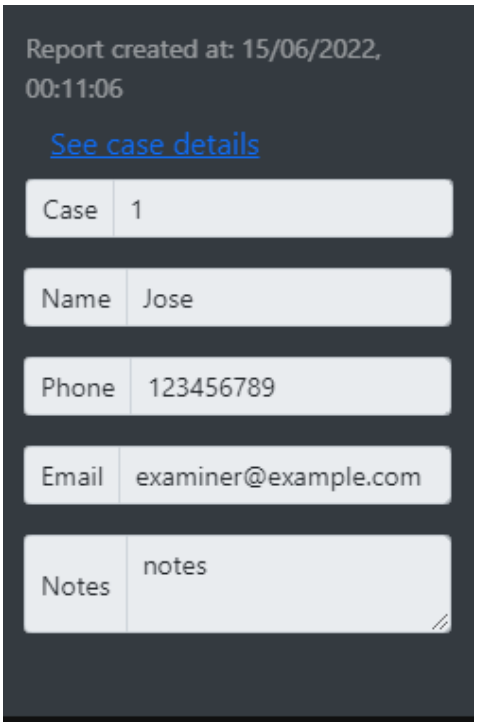

Figura 32: Detalhes do caso no módulo de report do MiFitAnalyzer

Separador Profiles. No separador inicial do módulo de report, intitulado de `Profiles` são revelados os utilizadores encontrados na aplicação. Nesta lista constam informações dos utilizadores que iniciaram sessão no dispositivo.

A Figura 33 mostra o separador `Profiles` do módulo de report com informações recolhidas dos utilizadores, onde são exemplos o nome, email, data de registo só para enumerar alguns.

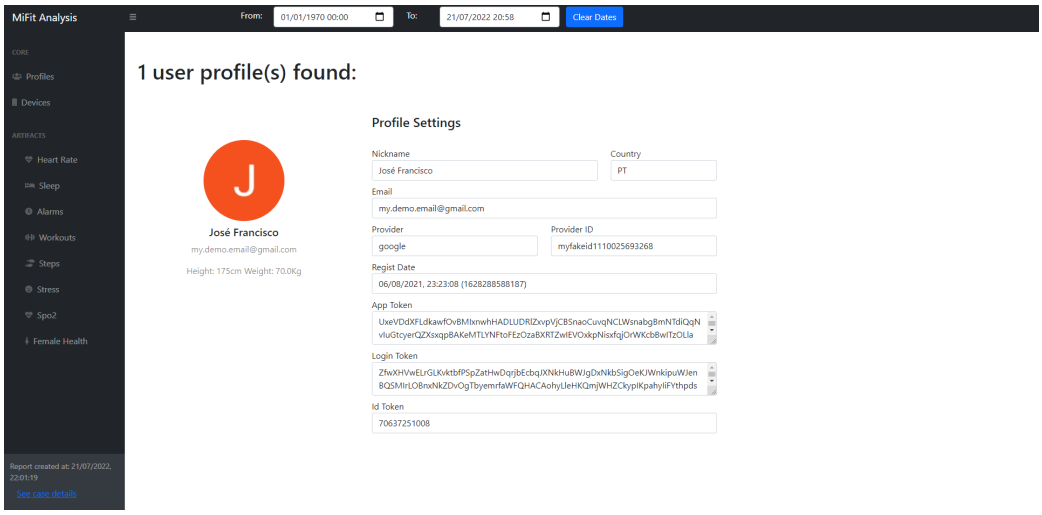

Figura 33: Separador Profiles do módulo de report do MiFitAnalyzer

Na eventualidade do utilizador utilizar um provedor de autenticação externo à MiFit (Google, Facebook ou WeChat), existe a possibilidade de extrair os tokens de acesso[17] a estes serviços de autenticação.

Tipicamente os tokens de acesso aos serviços alojados em cloud, onde se incluem o `App Token` e o `Login Token` apresentam um tempo de expiração curto. Não existe garantia que os tokens encontrados no dispositivo durante o processo de extração dos conteúdos do dispositivo ainda se encontrem validos à data da análise do relatório dinâmico.

Separador Alarms. Os ficheiros `alarms.html` e `alarms.js` renderizam o separador `Alarms` que contém uma tabela com alarmes programados pelo utilizador na pulseira MiBand. A coluna `time` corresponde à data e hora do alarme, a coluna `enabled` assume o valor `enable` quando o alarme está ativo, `disabled` quando desativado à data da captura dos conteúdos da aplicação MiFit.

A Figura 34 mostra o separador `Alarms` do relatório final dinâmico.

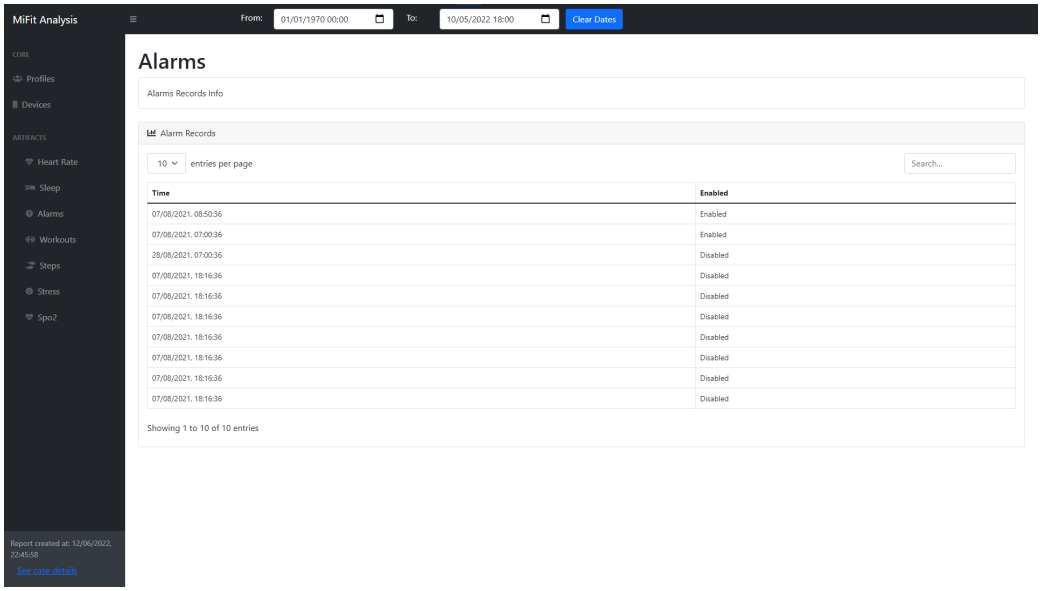

Figura 34: Separador de alarmes do módulo de report do MiFitAnalyzer

Separador Heart Rate. O artefacto representativo da captura dos batimentos cardíccos do utilizador é apresentado no separador `Heart Rate`.

A renderização deste separador é levada a cabo pelos ficheiros `heart-rate.html` e `heart-rate.js`. Como ilustrado na Figura 35, o separador `Heart Rate` exibe um gráfico de linhas com as leituras do batimento cardíaco medidas pela pulseira

17 Código de acesso que contém credenciais de segurança de um utilizador para uma sessão.

MiBand e mantidas na aplicação MiFit. No eixo das abcissas estão representados os instantes temporais das medições, nas ordenadas o valor do batimento cardiaco medido.

Todas as medições capturadas podem ser consultadas na tabela, associadas ao despositivo de medição (tipicamente MiBand).

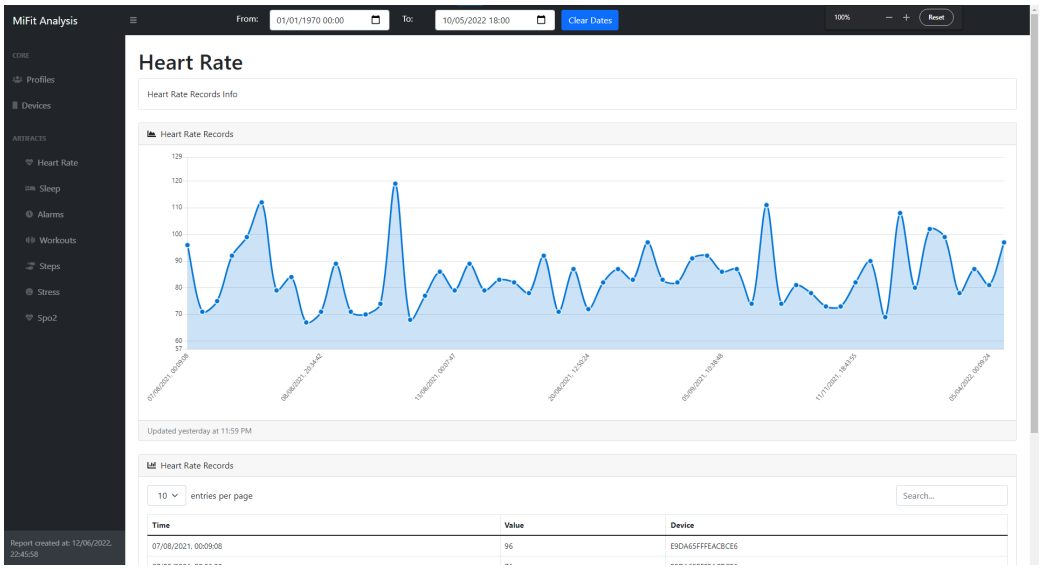

Figura 35: Separador Heart Rate do módulo de report do MiFitAnalyzer

Separador Sleep. Os ficheiros `sleep.html` e `sleep.js` reporta as informações do sono do utilizador.

No separador `Sleep` encontra-se a tabela que lista os registos de sono capturados, identificando o instante inicial e final associado à fase do sono (Figura 36).

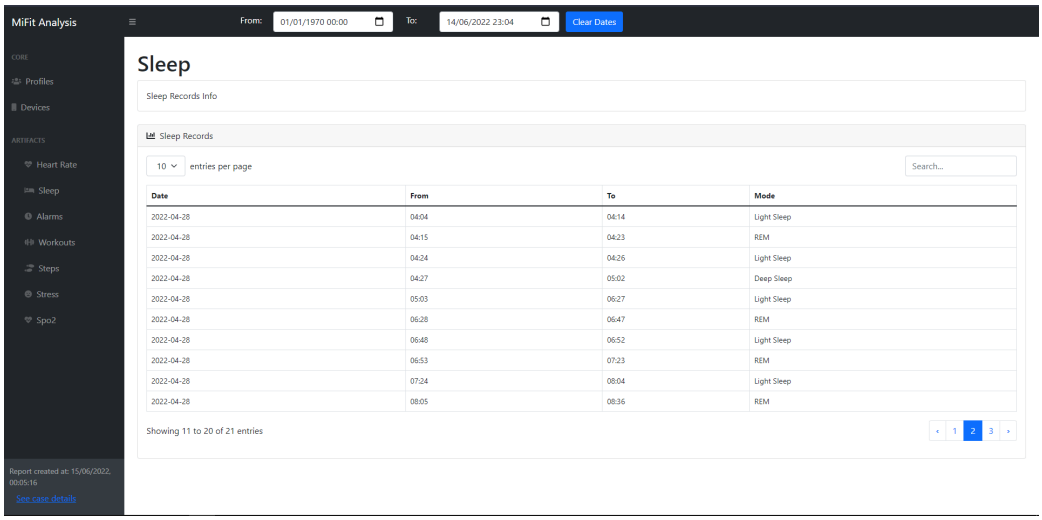

Figura 36: Separador Sleep do módulo de report do MiFitAnalyzer

Separador SPO2. A representação das medições da saturação de oxigénio (spo2) é feita pelos ficheiros `spo2.html` e `spo2.js`. Semelhante aos artefactos passiveis de serem representados numericamente (através de medições por exemplo), também o separador `Spo2` apresenta um gráfico de pontos onde é possível observar a variação nas medições.

A Figura 37 mostra o separador `Spo2` do relatório dinâmico da ferramenta MiFitAnalyzer.

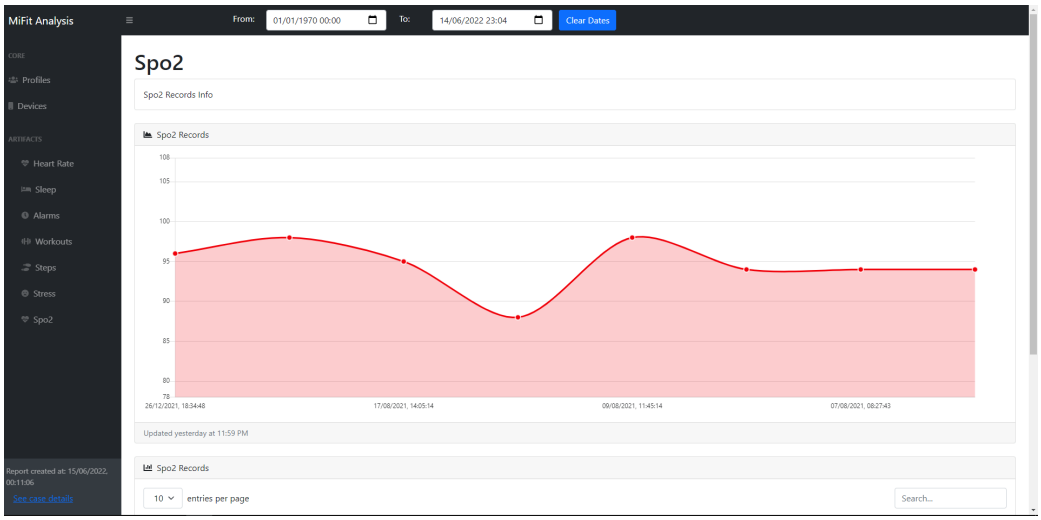

Figura 37: Separador Spo2 do módulo de report do MiFitAnalyzer

Separador Workouts. Para melhor entender entender o separador `Workouts`, seccionaremos em três partes. No topo da página (excluindo a barra de configuração das datas comum a todos os separadores) está representado um mapa. Na eventualidade de terem sido capturadas coordenadas GPS, a ferramenta MiFitAnalyzer encarregar-se-á de desenhar linhas representativas dos percursos efetuados pelo utilizador (Figura 38). Caso o perito forense pretenda ver as coordenadas num software externo, é possível exportar-las usando o botão Download Map.

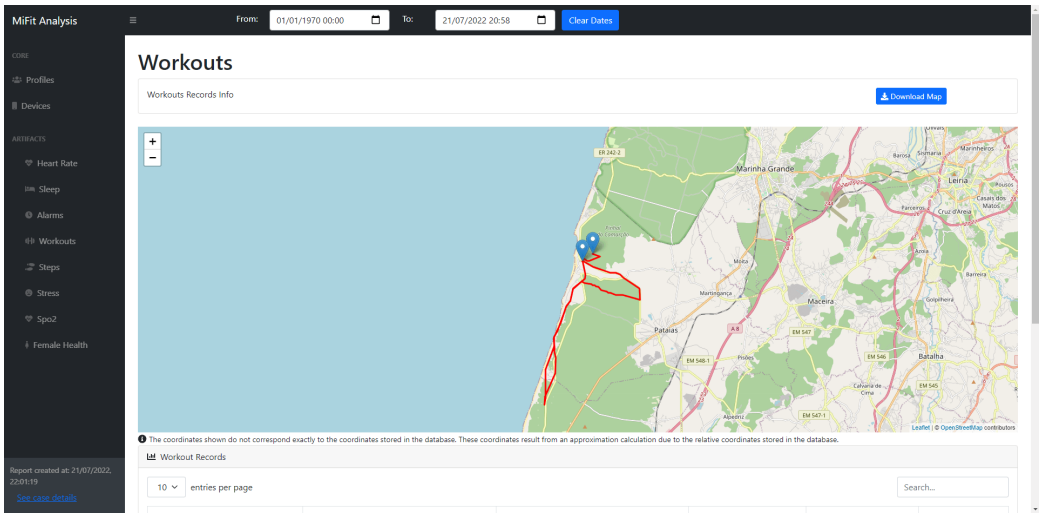

Figura 38: Mapa de Workouts do módulo de report do MiFitAnalyzer

Na segunda secção é representada uma tabela que lista as coordenadas GPS encontradas durante o período de análise e que estão associadas à pratica desportiva. Com objetivo de aumentar a usabilidade da aplicação, esta secção está colapsada por omissão, podendo ser mostrada pressionando o botão See Coordinates. Cada entrada na tabela representa uma coordenada encontrada, onde é possivel observar a sua latitude e longitude. O botão Add to map adiciona um marcador dinâmico ao mapa destacando assim as coordenadas identificadas na respetiva linha (Figura 39).

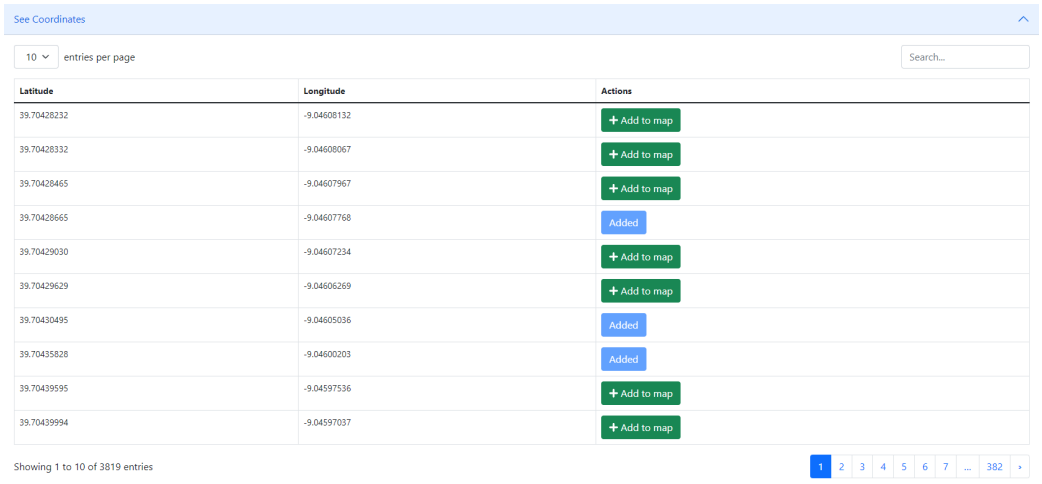

Figura 39: Tabela de coordenadas do separador Workouts do módulo de report do MiFitAnalyzer

As atividades desportivas encontradas são listadas na terceira secção, destacando o tipo de atividade, a data de início e fim da atividade, distância percorrida, calorias consumidas e o total de passos dados nessa atividade. De notar que nem todas as

atividades possuem todos os dados necessários para completar esta tabela. Exemplo disso é a atividade de natação ou alongamento, onde não existe, obviamente, a métrica "passos dados" durante qualquer um dessas atividades.

A Figura 40 mostra o a lista de atividades desportivas guardas na aplicação MiFit mostradas no separador `Workouts` no módulo de report na plataforma MiFitAnalyzer.

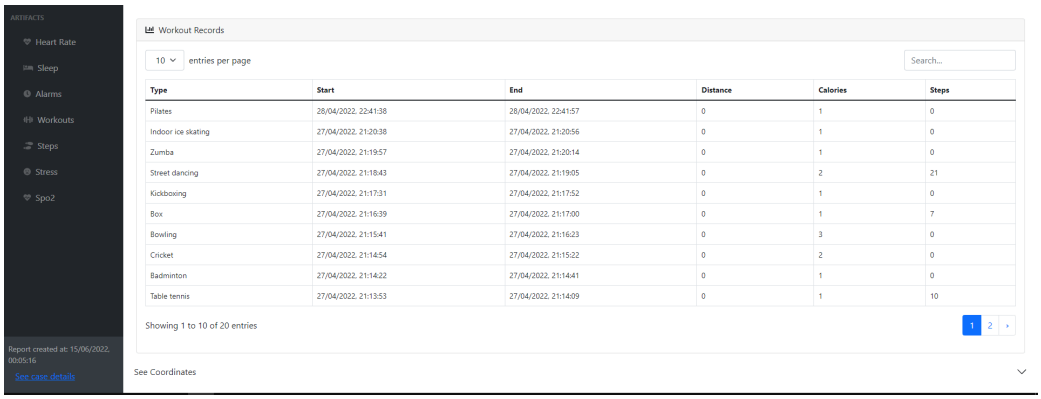

Figura 40: Tabela de Workouts do módulo de report do MiFitAnalyzer

Separador Devices. O separador `Devices` mostra os dispositivos (*wearables*) que estabeleceram ligação à aplicação MiFit. Para cada dispositivo apresentado, a plataforma MiFitAnalyzer descreve-e o produto e as suas principais funcionalidades. Da análise à aplicação MiFit, caso tenha sido possível extrair informação do dispositivo, esta será apresentada em forma de tabela.

Na Figura 41 é visível o separador `Devices`. Na análise exemplo observa-se uma pulseira MiBand 6 conectada a aplicação MiFit. O separador contém ainda informações que a identificam inequivocamente a pulseira inteligente conectada.

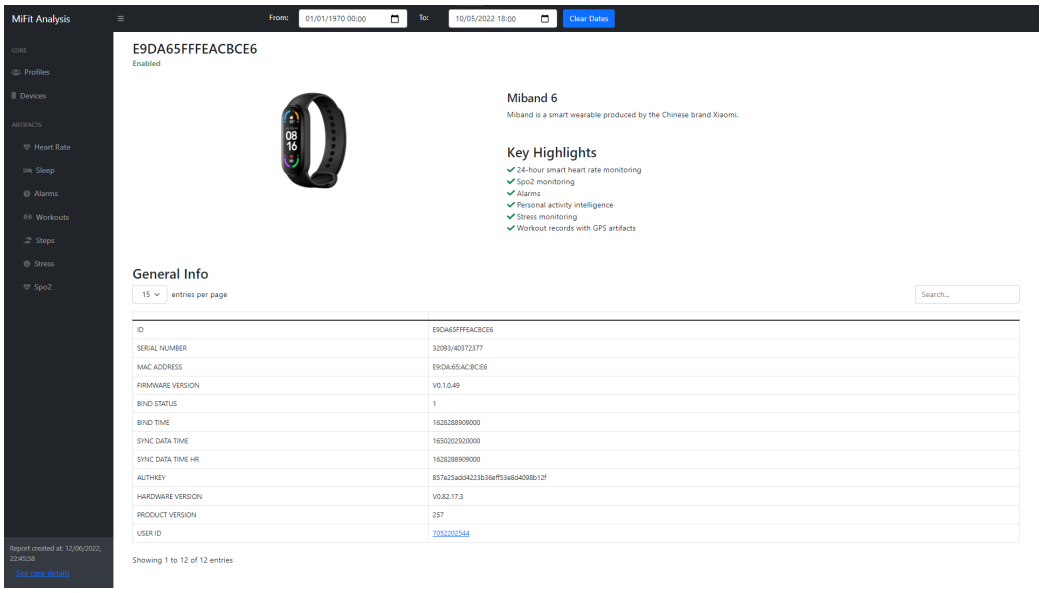

Figura 41: Separador Devices do módulo de report do MiFitAnalyzer

7.5 SUMÁRIO

Neste capítulo foi descrita a plataforma desenvolvida no âmbito deste projeto, a MiFitAnalyzer. A plataforma analisa os conteúdos da aplicação MiFit para Android, gerando relatórios ricos e dinâmicos recorrendo a um módulo de report também implementado neste projeto. Foram descritos os diferentes separadores deste módulo, ilustrando os artefactos forenses capturados durante o processo de análise.

Sendo o software Autopsy *open-source* e de referência para análise forense, foram feitas implementações que garantem a importação da ferramenta MiFitAnalyzer como módulos de Ingest e Report.

8

CONCLUSÕES

MiFit é uma aplicação usada para armazenar dados provenientes maioritariamente de *wearables* da marca chinesa Xiaomi.

Os dados são gerados por dispositivos conectáveis a aplicação MiFit, como é o caso das pulseiras inteligentes MiBand, vulgarmente chamadas *Fitness Trackers* capazes de definir e rastrear parâmetros de saúde e *fitness* durante o dia e à noite.

No âmbito deste projeto focou-se os esforços na busca de artefactos forenses num dispositivo Android primariamente provenientes da pulseira MiBand 6 conectada à aplicação MiFit. Para o efeito, foi criada uma plataforma que permite analisar conteúdos da aplicação MiFit, intitulada de MiFitAnalyzer.

Recorrendo à plataforma MiFitAnalyzer desenvolvida, é possível extrair artefactos importantes do ponto de vista forense, onde se inclui informações do utilizador, registos do batimento cardíaco, stress, padrões de sono, atividades desportivas e respetivas coordenadas GPS, alarmes definidos, passos dados, nível de saturação de oxigénio no sangue e informações dos dispositivos conectados.

Após todo o processamento dos dados, o formato de saída JSON facilita a leitura por outros softwares que possam a ler os conteúdos futuramente.

É ainda possível documentar todas as evidencias encontradas numa interface criada especialmente para o efeito, o `módulo de report`. Este módulo permite gerar relatórios dinâmicos recorrendo a tabelas, mapas, extração de coordenadas GPS, gráficos.

Sendo o Autopsy destacado como sendo uma das melhores ferramenta de análise forense em regime *open-souce* (Ingalls, 2021), foi garantido o suporte para esta ferramenta, atuando como um módulo de ingest externo.

Conclui-se então que o trabalho desenvolvido vão de encontro aos objetivos delineados no inicio do projeto.

8.1 TRABALHO FUTURO

Como implementações futuras, objetiva-se adicionar suporte para mais dispositivos que são passiveis de se conectar à aplicação MiFit.

Seria também interessante fazer uma análise dinâmica mais detalhada à MiFit, intercetando os pedidos entre a aplicação e os serviços *cloud*.

Sendo as comunicações bluetooth uma realidade como protocolo de comunicação de curto alcance, poderá ser interessante adicionar suporte para intercetar os pacotes entre a MiBand e a MiFit.

BIBLIOGRAFIA

Android Documentation (2020). *Android Runtime (ART) and Dalvik*. URL: https://source.android.com/devices/tech/dalvik.

Ceci, Laura (2022a). *Leading health and fitness apps in the Google Play Store worldwide in February 2022, by number of downloads*. URL: https://www.statista.com/statistics/690887/leading-google-play-health-worldwide-downloads/.

— (2022b). *Top fitness and Workout App Download growth 2022*. URL: https://www.statista.com/statistics/1239806/growth-top-fitness-mobile-apps-downloads/.

Collins, Bryan (mar. de 2020). *The Pomodoro Technique Explained*. URL: https://www.forbes.com/sites/bryancollinseurope/2020/03/03/the-pomodoro-technique/?sh=31459c7a3985.

Dougherty, Chad (jan. de 2009). *Vulnerability Note VU#836068 MD5 vulnerable to collision attacks*.

Ehringer, David (2010). «The dalvik virtual machine architecture». Em: *Techn. report (March 2010)* 4.8, p. 72.

Francisco, José et al. (2020). «Tiktok@Autopsy». Tese de doutoramento.

Hantke, Florian e Andreas Dewald (2020). «How can data from fitness trackers be obtained and analyzed with a forensic approach?» Em: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, pp. 500–508.

Henriksen, André et al. (2018). «Using fitness trackers and smartwatches to measure physical activity in research: analysis of consumer wrist-worn wearables». Em: *Journal of medical Internet research* 20.3, e9157.

Ingalls, Sam (Ago de 2021). *Best Digital Forensics Tools Software for 2022*. URL: https://www.esecurityplanet.com/products/digital-forensics-software/.

Kang, Serim, Soram Kim e Jongsung Kim (2020). «Forensic analysis for IoT fitness trackers and its application». Em: *Peer-to-Peer Networking and Applications* 13.2, pp. 564–573.

Laricchia, Federica (2022). *Wearables unit shipments worldwide from 2014 to 2021 (in millions), by vendor*. URL: https://www.statista.com/statistics/515634/wearables-shipments-worldwide-by-vendor/.

Nikishaev, Andrey (2018). *How I hacked my Xiaomi MiBand 2 fitness tracker — a step-by-step Linux guide*. URL: https://medium.com/machine-learning-world/how-i-hacked-xiaomi-miband-2-to-control-it-from-linux-a5bd2f36d3ad.

Oh, G (2021). *Export mi band 5 data from Mi Fit to Google Sheets*. URL: https://dev.to/neuronmachine/export-mi-fit-data-to-google-sheets-2ed4.

Paradiso, Chloe, Franciscio Colino e Sam Liu (2020). «The validity and reliability of the mi band wearable device for measuring steps and heart rate». Em: *International journal of exercise science* 13.4, p. 689.

Sandvik, Bjørn (2008). «Using KML for thematic mapping». Em: *Institute of Geography School of GeoSciences. Edinburgh, University of Edinburgh. MSc in Geographical Information Science* 22.

Sathe, Sneha C e Nilima M Dongre (2018). «Data acquisition techniques in mobile forensics». Em: *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 280–286. DOI: 10.1109/ICISC.2018.8399079.

Sawh, Michael (2021). *Xiaomi Mi Band 6 v Mi Band 5: the key differences*. URL: https://www.wareable.com/fitness-trackers/xiaomi-mi-band-6-vs-mi-band-5-8382.

Shirke, kshitija (jan. de 2019). URL: https://medium.com/@kshitishirke/mobile-security-framework-mobsf-static-analysi-df22fcdae46e.

Statista (2021). *Leading health and fitness apps in the Google Play Store worldwide in October 2021*. URL: https://www.statista.com/statistics/690887/leading-google-play-health-worldwide-downloads/.

Wei, Xuetao et al. (2012). «Permission evolution in the Android ecosystem». Em: *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 31–40.

World Health Organization (dez. de 2021). *Oxygen*. URL: https://www.who.int/news-room/questions-and-answers/item/oxygen.

Xie, Junqing et al. (abr. de 2018). «Evaluating the Validity of Current Mainstream Wearable Devices in Fitness Tracking Under Various Physical Activities: Comparative Study». Em: *JMIR Mhealth Uhealth* 6.4. ISSN: 2291-5222. DOI: 10.2196/mhealth.9754. URL: http://www.ncbi.nlm.nih.gov/pubmed/29650506.

Zisko, Nina et al. (2017). «Personal Activity Intelligence (PAI), sedentary behavior and cardiovascular risk factor clustering–the HUNT study». Em: *Progress in Cardiovascular Diseases* 60.1, pp. 89–95.

D E C L A R A Ç Ã O

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"Análise forense digital à aplicação móvel MiFit"*, é original e foi realizado por José Carlos Ferreira Francisco (2202274) sob orientação do Professor Doutor Miguel Monteiro de Sousa Frade e do Professor Doutor Patrício Rodrigues Domingues.

Leiria, Setembro de 2022

Assinado por: **JOSÉ CARLOS FERREIRA FRANCISCO**
Num. de Identificação: 15612610
Data: 2022.09.28 23:50:57+01'00'

José Carlos Ferreira Francisco