

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Patient Generated Health Data By Glintt

João Luís Maia Gomes



Mestrado em Engenharia de Software

Supervisor: Professor Nuno Flores

Supervisor da Empresa: Engenheiro João Gomes

16 de Julho de 2018

Patient Generated Health Data By Glintt

João Luís Maia Gomes

Mestrado em Engenharia de Software

Resumo

Com o número de *wearables* a aumentar nos últimos anos, nasceu uma forma fácil e eficaz de recolha de dados, criando um aglomerado de informação que tenta melhorar o estilo de vida dos utilizadores. Para além dos *wearables* se focarem em melhorar o estilo de vida, começam a surgir alguns mais focados em melhorar aspetos de saúde dos utilizadores. Com tanta informação a ser explorada apenas pelos fabricantes dos *wearables* através de aplicações que interagem com os utilizadores que os adquiram, é deixada de parte uma importante fatia de utilizadores que realmente podem tirar importantes conclusões: os agentes de saúde (médicos, pediatras, fisioterapeutas, entre outros). Para explorar esta oportunidade, nasceu um conceito denominado por *Patient Generated Health Data (PGHD)*, que defende que os dados gerados fora do ambiente clínico também devem ser tratados e explorados por agentes de saúde. Sendo assim, com a informação recolhida pelos *wearables* (gerada fora de um ambiente clínico) surge uma oportunidade de negócio com grande potencial.

Torna-se, portanto, pertinente, abordar a problemática em torno da heterogeneidade da informação gerada pelos vários *wearables*. Para tal, é conveniente o desenvolvimento de um sistema que permita provar que é possível agregar dados de várias fontes de dados (fornecedores de *wearables*) num modelo unificado (único) de dados, sendo estes visualizados numa aplicação móvel fornecida ao paciente. Para além do desenvolvimento de um sistema é também importante criar um sistema modular, que permita incorporar novos componentes para que os dados possam ser visualizados pelo agente de saúde, podendo analisar informação vital do utilizador e retornar um conjunto de avaliações importantes relativas à saúde do utente.

Para resolver este problema, foi desenvolvido um sistema de software que comunica com duas plataformas de fontes de dados (*Fitbit* e *Google Fit*), em que os dados são convertidos num modelo comum que, por sua vez, é visualizado numa aplicação móvel. Esta solução propôs-se a atingir diversos objetivos, nomeadamente: definição do modelo unificado de dados, desenvolvimento de componentes de comunicação com as fontes de dados, desenvolvimento da aplicação móvel, análise das fontes de dados e respetivo caso de estudo de validação.

Atingidos os objetivos propostos, torna-se possível a monitorização de pacientes independentemente do local em que estes se encontram, aplicável a quaisquer serviços de recolha de dados de saúde.

Palavras-chave: *Patient Generated Health Data*, *Wearables*, Aplicação Móvel, *Google Fit*, *Fitbit*, Modelo Unificado de Dados.

Abstract

With the number of wearables increasing in recent years, an easy and effective way of collecting data was born, creating a cluster of information that attempts to improve the lifestyle of users. Besides, the wearables focus on improving their lifestyle, new wearables focused on improving the health of users are born. With so much information being exploited only by the wearable creators, through applications that interact with the users who acquire them, a large share of users are left aside, who can really draw important conclusions: health workers (doctors, pediatricians, physiotherapists, among others). To explore this opportunity, a concept called Patient Generated Health Data (PGHD) was born, which believes that data generated outside the clinical environment should also be treated and explored by health workers. Thus, with the information collected by the wearables (generated outside of a clinical environment) a business opportunity with great potential arises.

The problem that this project proposes to resolve is a system that allows to prove that it is possible to aggregate data from several data sources (wearable suppliers) into a unified (single) data model, which that data is visualized in a application provided to the patient. Thus, it is possible to create a system that can prove that, easily, another component can be incorporated to show that data to the health agents, enable them to analyze vital information of the user to return important evaluation relative to user's health.

To solve this problem, a software system has been developed that communicates with two platforms of data sources (Fitbit and Google Fit). The data is converted into a common data model which works as source of data to a mobile application. This solution can be decomposed in several goals: unified data model, developing communication components to get information from data sources, mobile application, study of data sources, case study of a new data source, among others.

With the proposed goals reached, a new way which makes possible to monitor patients regardless of where they are located is born. In conclusion, all proposed goals have been achieved, creating new challenges that serve as future work.

Keywords: Patient Generated Health Data, Wearables, Mobile Application, *Google Fit*, *Fitbit*, Unified Data Model.

Agradecimentos

Em primeiro lugar tenho de agradecer aos meus pais, pelo constante apoio nos últimos cinco anos desta minha jornada académica.

Em segundo lugar, um grande obrigado para o meu orientador da FEUP e o supervisor da empresa (respetivamente professor Nuno Flores e Engenheiro João Gomes) por todo o apoio e *feedback* dado durante esta dissertação.

Para terminar, também gostaria de agradecer aos restantes elementos da minha família e amigos. Juntamente, com todos os colegas da Faculdade de Engenharia da Universidade do Porto, professores do mestrado em Engenharia de Software e a equipa da *Glintt*.

João Gomes

*“Approach everything with an open mind, with a learning mind.
You will never stop learning as long as you keep the mind-set that everything works, because
everything does work.
There’s a time and a place for every single move.
If you work on it enough, it will work.”*

Conor McGregor

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação e Objetivos	1
1.3	Organização do documento	2
2	Wearables	5
2.1	O que é um <i>wearable</i>	5
2.2	Mercado <i>Wearable</i>	6
2.2.1	<i>Google Trends</i>	7
2.2.2	Aparelhos ligados à Internet e valor de mercado	7
2.2.3	<i>Wearables</i> por Segmento	7
2.3	Contextos de aplicação	8
2.3.1	Caminhada	10
2.3.2	Sono	10
2.3.3	Diabetes	10
2.3.4	Postura	10
2.3.5	Risco de queda	11
2.4	Classificação Legal de <i>Wearables</i> nos EUA	11
2.4.1	Aparelho Médico	11
2.4.2	Modelo de classificação	12
2.5	Classificação Legal de <i>Wearables</i> na UE	14
2.5.1	Aparelhos Médicos	14
2.5.2	Modelo de classificação	15
2.6	Sumário	15
3	Dados de saúde	17
3.1	Tríade CIA	17
3.1.1	Confidencialidade	17
3.1.2	Integridade	18
3.1.3	Disponibilidade, em inglês <i>Availability</i>	18
3.2	Standards de segurança	18
3.2.1	ISO/IEC 27002:2013	19
3.2.2	ISO 27799:2016	20
3.3	Regulação Geral da Proteção de Dados (RGPD)	20
3.3.1	Tratamento de dados de forma segura	20
3.3.2	Transmissão dos dados de uma forma segura	24
3.4	Sumário	26

4	<i>Patient Generated Health Data com Wearables</i>	27
4.1	<i>Patient Generated Health Data (PGHD)</i>	27
4.1.1	Definição	28
4.1.2	Vantagens	29
4.1.3	Obstáculos	30
4.2	Desafios	31
4.3	Trabalho relacionado	31
4.4	Fator diferenciador da solução	32
4.5	Sumário	33
5	Âmbito da solução	35
5.1	Escolha do(s) contexto(s) de aplicação	35
5.2	Plataformas de fontes de dados	38
5.2.1	<i>Google Fit</i>	39
5.2.2	<i>Fitbit</i>	41
5.2.3	<i>Apple Healthkit</i>	41
5.2.4	<i>Tizen/Samsung</i>	42
5.2.5	Escolha das plataformas	42
5.3	Escolha do <i>wearable</i>	42
5.4	RGPD	44
5.5	Standards de segurança	45
6	Desenho da solução	47
6.1	Estudo ao modelo de dados das fontes escolhidas	47
6.1.1	<i>Google Fit</i>	48
6.1.2	<i>Fitbit</i>	50
6.2	Modelo de dados	51
6.2.1	<i>Utilizador</i>	52
6.2.2	<i>Valores corporais</i>	54
6.2.3	<i>Batimento cardíaco</i>	54
6.2.4	<i>Exercício físico</i>	54
6.2.5	<i>Sono</i>	55
6.2.6	Notas finais	56
6.3	Caso de estudo - <i>Samsung Health Server SDK</i>	58
6.3.1	Alterações à base de dados	59
6.3.2	Controlador - <i>SDictionary</i>	60
6.4	Possíveis soluções para reduzir o impacto de alterações na base de dados	60
6.5	Solução tecnológica	62
7	Metodologia de desenvolvimento	63
7.1	Metodologia de desenvolvimento de software	63
7.2	Planeamento	64
7.3	Reuniões	65
8	Desenvolvimento da solução	67
8.1	Linguagens de Programação e <i>Frameworks</i>	67
8.1.1	<i>React Native</i>	67
8.1.2	<i>Node.js</i> e <i>MySQL</i>	68
8.2	Requisitos	70

8.2.1	Funcionais	70
8.2.2	Não-Funcionais	70
8.3	Arquitetura	72
8.4	<i>Design</i>	73
8.4.1	Componentes da aplicação móvel	73
8.4.2	Fluxo comunicação entre API e Aplicação Móvel	73
8.4.3	Fluxo comunicação entre as fontes de dados e o sistema	76
8.5	Testes	78
8.6	Validação	79
8.6.1	Execução dos componentes <i>FDictionary</i> e <i>GDictionary</i>	80
8.6.2	Resultado na base de dados	80
8.6.3	Aplicação móvel	82
8.6.4	Fonte de dados	83
9	Conclusão	85
9.1	Contribuições	85
9.2	Dificuldades	86
9.3	Trabalho Futuro	86
A	Modelo de dados das fontes	87
A.1	Modelo de dados da <i>Google Fit</i>	87
A.2	Modelo de dados da <i>Fitbit</i>	87
B	Resultado dos testes Unitários	93
C	Ecrãs da aplicação móvel	95
D	<i>Print-Screens</i> da base de dados	99
E	Fontes de dados <i>print-screens</i>	101
	Referências	103

Lista de Figuras

2.1	<i>Fitbit Charge 2</i>	6
2.2	<i>Apple Smartwatch</i>	6
2.3	<i>Dexcom G5</i>	6
2.4	<i>Alex Posture</i>	6
2.5	<i>Google Trends em Wearables</i>	7
2.6	Número de aparelhos ligados à Internet, numa escala mundial desde 2016 até 2021 (em milhões)	8
2.7	Valor de mercado de aparelhos <i>wearable</i> desde 2010 até 2018 (em milhões de dólares americanos)	8
2.8	Vendas do Mercado <i>Wearable</i> mundialmente, por segmento, em 2018	9
2.9	Modelo de classificação da FDA	13
3.1	DPIA explicado	22
3.2	Processo de DPIA	23
4.1	PGHD fluxo de dados	28
5.1	Quota de mercado de fornecedores de <i>wearables</i> entre 2014 e 2017	37
5.2	Aplicações de saúde/ <i>fitness</i> mais populares nos EUA (com número de utilizadores ativos por mês) desde Julho de 2017	37
5.3	Estágios de sono na aplicação da <i>Fitbit</i>	39
5.4	<i>Google Fit</i> arquitetura de alto nível	41
5.5	<i>Blaze Fitbit</i>	43
5.6	<i>Ionic Fitbit</i>	43
5.7	<i>Charge 2 Fitbit</i>	43
6.1	Conceito de sessão na <i>Google Fit</i>	49
6.2	1 para muitos	52
6.3	1 para 1	52
6.4	Modelo de dados - Utilizador	53
6.5	Modelo de dados - Valores corporais	54
6.6	Modelo de dados - Batimento Cardíaco	55
6.7	Modelo de dados - Exercício Físico	56
6.8	Modelo de dados - Sono	57
6.9	Arquitetura de alto nível	62
7.1	Planeamento primeira fase	65
7.2	Planeamento segunda fase	65

8.1	Arquitetura do <i>Node.js</i>	69
8.2	Diagrama de componentes	72
8.3	Diagrama de componentes da aplicação móvel	74
8.4	<i>REQ-FUNC1</i> Diagrama de Sequência	74
8.5	<i>REQ-FUNC2</i> Diagrama de Sequência	75
8.6	<i>REQ-FUNC3</i> Diagrama de Sequência	75
8.7	<i>REQ-FUNC4</i> Diagrama de Sequência	76
8.8	<i>REQ-FUNC5</i> Diagrama de Sequência	76
8.9	Execução do <i>script</i> do componente <i>FDictionary</i>	80
8.10	Execução do <i>script</i> do componente <i>GDictionary</i>	80
8.11	Tabela <i>User</i>	81
8.12	Tabela <i>BodyValues</i>	81
8.13	Tabela <i>Activity</i> parte 1	81
8.14	Tabela <i>Activity</i> parte 2	82
8.15	Ecrã da funcionalidade <i>REQ-FUNC1</i>	82
8.16	Ecrã da funcionalidade <i>REQ-FUNC2</i>	82
8.17	Ecrã da funcionalidade <i>REQ-FUNC4</i>	82
8.18	Informação do utilizador na <i>Fitbit</i>	83
8.19	Informação dos valores corporais na <i>Fitbit</i>	83
8.20	Informação dos exercícios físicos corporais na <i>Fitbit</i>	83
8.21	Exercício número 1 da <i>Google Fit</i>	84
8.22	Exercício número 2 da <i>Google Fit</i>	84
8.23	Informação dos valores corporais na <i>Google Fit</i>	84
A.1	Modelo de dados <i>Google Fit</i>	88
A.2	Modelo de dados <i>Fitbit</i> - Atividade	89
A.3	Modelo de dados <i>Fitbit</i> - Batimento cardíaco	90
A.4	Modelo de dados <i>Fitbit</i> - Corpo e Peso	90
A.5	Modelo de dados <i>Fitbit</i> - Sono	91
B.1	Resultado dos testes unitários	93
C.1	Ecrã da funcionalidade <i>REQ-FUNC4</i>	96
C.2	Ecrã da funcionalidade <i>REQ-FUNC5</i>	97
D.1	Tabela <i>Userconfigurationfitbit</i>	99
D.2	Tabela <i>Userconfigurationgooglefit</i>	99
D.3	Tabela <i>Device</i>	99
D.4	Tabela <i>Heartratevalue</i>	99
D.5	Tabela <i>Heartratezone</i>	99
D.6	Tabela <i>Sleep</i>	100
D.7	Tabela <i>SleepData</i>	100
D.8	Tabela <i>SleepTypeOfReading</i>	100
E.1	Informação de sono da <i>Fitbit</i>	101
E.2	Informação do utilizador da <i>Google Fit</i>	101

Lista de Tabelas

6.1	Comparação de informação do modelo de dados entre fontes	56
8.1	Métodos HTTP da fonte <i>Fitbit</i>	77
8.2	Métodos HTTP da fonte <i>Google Fit</i>	78

Acrónimos

API	<i>Application Programming Interface</i>
CE	Comissão Europeia
CIA	<i>Confidentially Integrity Availability</i> (em português Confidencialidade, Integridade e Disponibilidade)
CSV	<i>Comma-separated values</i>
DPIA	Avaliação de Impacto de Proteção de Dados
EHR	<i>Electronic Health Record</i>
EUA	Estados Unidos da América
FDA	<i>Food and Drug Administration</i>
FEUP	Faculdade de Engenharia da Universidade do Porto
GPS	Sistema de Posicionamento Global
HDE	Isenção de Dispositivos Humanitários
HDU	Designação do Dispositivo de Uso Humanitário
HTTP	<i>HyperText Transfer Protocol</i>
IVD	Dispositivos de Diagnóstico <i>In-Vitro</i>
JSON	<i>JavaScript Object Notation</i>
ONC	<i>Office of the National Coordinator for Health Information Technology</i>
PGHD	<i>Patient Generated Health Data</i>
PMA	Aprovação Pré-Comercial
REST	<i>Representational State Transfer</i>
RGPD	Regulamentação Geral de Proteção de Dados
UE	União Europeia
XML	<i>eXtensible Markup Language</i>

Capítulo 1

Introdução

Este capítulo é a base introdutória ao projeto e dividi-se em 3 secções:

- Secção 1.1: faz o enquadramento do projeto, onde foi desenvolvido e em que contexto foi feito.
- Secção 1.2: justifica o que motivou o desenvolvimento do projeto, bem como os objetivos pretendidos a atingir.
- Secção 1.3: apresenta a estrutura dos próximos capítulos.

1.1 Enquadramento

O projeto foi desenvolvido no âmbito da unidade curricular de Dissertação [1] do segundo ano do Mestrado em Engenharia de Software (MESW) [2] da Faculdade de Engenharia da Universidade da Porto (FEUP) [3]. A proposta da dissertação surgiu pela *Glintt* [4] e o desenvolvimento decorreu nos escritórios localizados no Porto.

A *Glintt* é uma empresa com mais de 20 anos de experiência, focados na consultoria e serviços tecnológicos na área da saúde, sendo que muitas das suas soluções são utilizadas em mais de 200 hospitais e clínicas. No setor da farmácia, mais de 12000 farmácias na Península Ibérica utilizam o software de gestão suportado pela *Glintt*.

1.2 Motivação e Objetivos

Este projeto surgiu no contexto do novo programa de mobilidade da *Glintt*, em que se pretende introduzir novas soluções sem-fios e móveis para serem incorporadas nos serviços já existentes. Portanto, com o vasto conhecimento da empresa na área de software da saúde e com os *wearables* a se afirmarem no mercado como uma força dominante, nasceu um oportunidade para expandir o mercado para além das soluções de software que já estavam a ser desenvolvidas pela empresa. Sendo assim, foram identificadas pela empresa cinco áreas (*Caminhada, Sono, Postura, Diabetes*

e *Risco de queda*) com muito potencial a serem exploradas, sendo que o próximo passo é identificar, nestas áreas, *wearables* que consigam recolher este tipo de informação. Com a fonte de dados (*wearables*) identificada, falta só perceber como este projeto se enquadra na área de saúde. Os *wearables* permitem de uma forma simples e rápida a recolha de informação vital do utilizador, mas parte desta informação está a ser lida e apresentada aos utilizadores que adquiriram os aparelhos, isto é, esta informação não está a ser explorada de acordo com o seu potencial.

Para isso, é introduzido outro conceito (que é considerado o conceito principal do projeto) denominado por *Patient Generated Health Data*, que afirma que a informação dos utentes gerada fora do ambiente clínico, também deve ser explorada por agentes de saúde. Em conjunto com os *wearables* como fontes de dados, o objetivo principal é desenvolver uma solução que irá aproveitar a informação capturada por este tipo de aparelhos de diferentes fornecedores. Dentro desse objetivo principal, é necessário desenvolver um modelo unificado de dados que junta essa informação. Para além do modelo comum de dados, também é necessário implementar uma aplicação móvel que tratará essa informação apresentando-a ao utente. Antes de terminar esta secção, é crucial entender que este projeto assenta na perspetiva do utente, sendo que o papel do agente de saúde não faz parte do contexto desta dissertação.

Para terminar, o objetivo principal deste projeto é desenvolver um sistema de software que permita aproveitar dados gerados por utentes (sem que estes estejam presentes num ambiente clínico), através do uso de *wearables* como fontes de recolha de dados. Devido à complexidade deste objetivo foram definidos os seguintes sub-objetivos:

- Estudo do modelo de dados de cada plataforma de fonte de dados (secção 6.1);
- Conceção de um modelo de dados que permita agregar informação de diferentes fornecedores de *wearables* (secção 6.2);
- Desenvolvimento de um sistema de software, que tem por base uma aplicação móvel que interage com os utentes (capítulo 8);
- Caso de estudo de inserção de uma nova plataforma de fonte de dados, que serve como validação do modelo unificado de dados (secção 6.3).

1.3 Organização do documento

A dissertação está dividida em 8 capítulos principais:

- Capítulo 2: Contextualiza os aspetos referentes à tecnologia *wearable*.
- Capítulo 3: Aborda questões relativas à segurança de informação de dados de saúde.
- Capítulo 4: Define o problema de forma sucinta e clara, justifica a sua relevância, apresenta soluções já existentes para o resolver e justifica em que a solução implementada se destaca em relação à concorrência.

- Capítulo 5: Apresenta as decisões de alto nível referentes à solução proposta para a problema identificado.
- Capítulo 6: Apresenta as várias decisões que levam ao modelo de dados, prova a qualidade do mesmo e apresenta a solução tecnológica.
- Capítulo 7: Explica a metodologia e o planeamento usado no projeto.
- Capítulo 8: Apresenta e explica aspetos da implementação da solução.
- Capítulo 9: É feito um balanço final às contribuições, objetivos propostos, dificuldades encontradas e ao trabalho futuro.

Capítulo 2

Wearables

Neste capítulo é abordado um dos temas centrais deste projeto: *Wearables*. Por sua vez, encontra-se dividido nas seguintes secções:

- 2.1: define de uma forma sucinta e clara o que pode ser considerado um aparelho *wearable*.
- 2.2: justifica a importância deste tema, através de estatísticas relativas à evolução do mercado dos *wearables*.
- 2.3: apresenta os contextos em que os *wearables* atuam.
- 2.4: explica como é feito o processo legal de classificação de *wearables* nos EUA.
- 2.5: explica como é feito o processo legal de classificação de *wearables* na UE.
- 2.6: permite reter os principais tópicos deste capítulo.

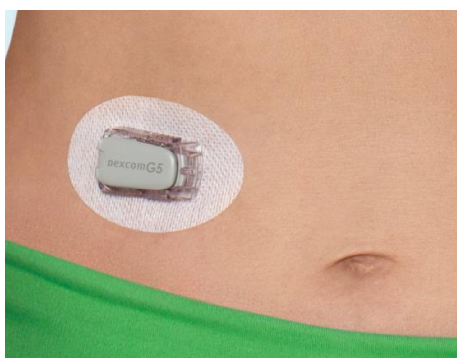
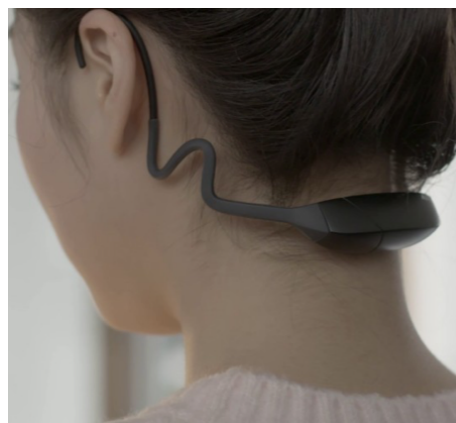
2.1 O que é um *wearable*

A palavra *wearable* pode ser decomposta em *wear* e *able*: algo que pode ser vestido. O dicionário de Oxford define *wearable* como "um item que pode ser vestido", "fácil de vestir" e "adequado para vestir" [5]. Sendo assim, na perspectiva de engenharia de software e de acordo com o dicionário de Oxford, *wearable* é definido como "um computador ou outro aparelho eletrónico que é pequeno e leve, e facilmente vestido ou carregado no corpo de um indivíduo" [5].

Os tipos mais comuns de *wearables* são as pulseiras inteligentes e os *smartwatches*, retratados nas figuras 2.1 [6] e 2.2 [7] respetivamente. *Smartwatches* são aparelhos que podem ser comparados a pequenos computadores, tem aplicações, leem dados GPS e batimento cardíaco, entre outros. As pulseiras inteligentes são parecidas com os *smartwatches*, mas tem o poder computacional menor, podem na mesma ler GPS e batimento cardíaco. Ambos podem ser facilmente usados como peça de vestuário ou indumentária e são compostos por vários sensores que conseguem extrair uma variedade de informação.

Figura 2.1: *Fitbit Charge 2*Figura 2.2: *Apple Smartwatch*

Também existem outros aparelhos como o *Dexcom G5* (figura 2.3 [8]) que lê os níveis de glucose através de sensores em contacto com a pele. Outro exemplo com um design menos comum é o *Alex Posture* (figura 2.4 [9]), que utiliza sensores para avaliar a postura corporal.

Figura 2.3: *Dexcom G5*Figura 2.4: *Alex Posture*

Embora todos os *wearables* adotem diferentes abordagens de design, todos têm um objetivo em comum: design amigável, ou seja, a sua conceção fomenta uma interação clara, fácil e intuitiva com o utilizador.

2.2 Mercado *Wearable*

Nesta secção são apresentadas estatísticas consideradas relevantes sobre a evolução do mercado dos *wearables* nos últimos anos. Com as estatísticas apresentadas nas secções 2.2.1 [10], 2.2.2 [11] e 2.2.3 [12], tem-se notado que recentemente este mercado tem ganhado uma grande atenção no interesse dos utilizadores, fazendo com que esta área cresça de uma forma exponencial. Numa primeira fase, é abordado a evolução da procura da palavra *wearables* no motor de pesquisa da *Google*. De seguida, são apresentadas algumas métricas que justificam o interesse crescente

nesta área, como o número de aparelhos utilizados numa escala global e o valor do mercado. Por fim, é apresentado o valor de vendas, juntamente como estas se encontram segmentadas.

2.2.1 Google Trends



Figura 2.5: *Google Trends* em *Wearables*

De acordo com o *Google Trends*¹, desde 2012 até 2015 a evolução é grande, com os *wearables* a terem um aumento enorme no foco dos utilizadores (figura 2.5 [10]). É importante ainda referir que na linha horizontal do gráfico da figura 2.5 a unidade é o nível de popularidade. Em que um valor de 100 é o pico de popularidade do termo, um valor de 50 significa que o termo teve metade da popularidade e uma pontuação de 0 significa que não houve dados suficientes para classificar a palavra.

2.2.2 Aparelhos ligados à Internet e valor de mercado

*Statista*² estima que o número de *wearables* triplique desde 2016 (325 milhões) até 2021 (929 milhões), como mostra a figura 2.6 [11]. Se o número de *wearables* é um forte indicador, o valor de mercado também o é. Consequentemente, *Statista* (figura 2.7 [13]) mostra que desde 2012 até 2018 o valor de mercado irá aumentar por um fator de 10 (1,260 milhões de dólares até 12,642 milhões de dólares).

2.2.3 *Wearables* por Segmento

As figuras 2.5, 2.6 e 2.7 analisam o impacto dos *wearables* no geral, por outro lado a figura 2.8 [12] tenta perceber como esta área se encontra segmentada. É possível perceber que o grande impacto deste conceito está focado no estilo de vida, ou seja, como e de que modo é possível beneficiar e melhorar o estilo de vida dos utilizadores. É de referir, que para além de melhorar o estilo de vida dos utilizadores, os *wearables* tentam servir de forma direta profissionais de saúde como importantes ferramentas de auxílio. Para concluir, pode-se afirmar que o principal propósito dos *wearables* é dar ao utilizador um dispositivo de dimensões reduzidas, poderoso e fácil de usar para melhorar a sua saúde e o seu estilo de vida.

¹Ferramenta que mostra os conceitos mais populares num passado recente no motor de pesquisa da *Google*.

²Web-site que permite aceder a estatísticas referentes a vários temas, disponível em <https://www.statista.com/>.

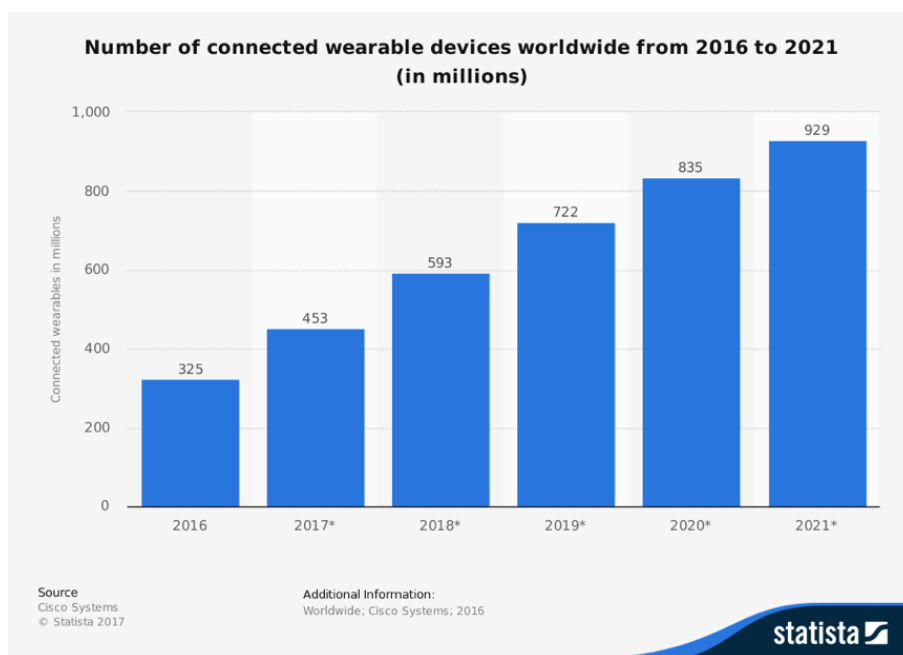


Figura 2.6: Número de aparelhos ligados à Internet, numa escala mundial desde 2016 até 2021 (em milhões)

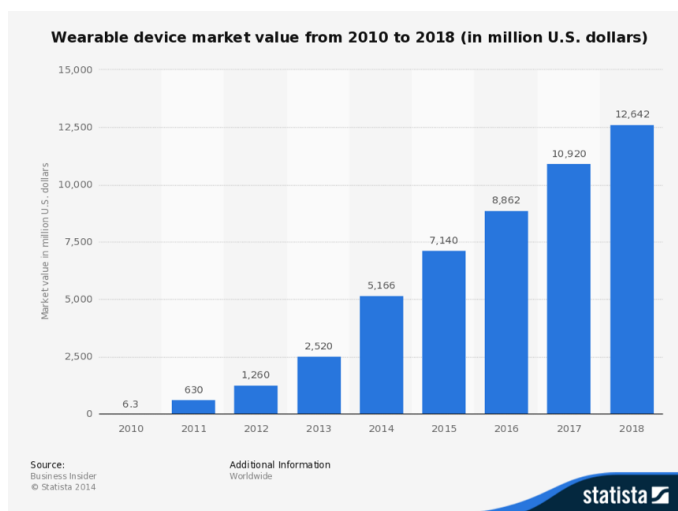


Figura 2.7: Valor de mercado de aparelhos *wearable* desde 2010 até 2018 (em milhões de dólares americanos)

2.3 Contextos de aplicação

O bem-estar dos utilizadores é um aspeto que os *wearables* abordam, fazendo com que áreas como *Caminhada* e *Sono* sejam consideradas o foco principal da tecnologia *wearable*, existindo um conjunto grande de fornecedores como *Fitbit* [14], *Garmin* [15], *Google* [16], *Apple* [17], *Samsung* [18], entre outros. Outra área que se foca no bem-estar dos utilizadores e que começa

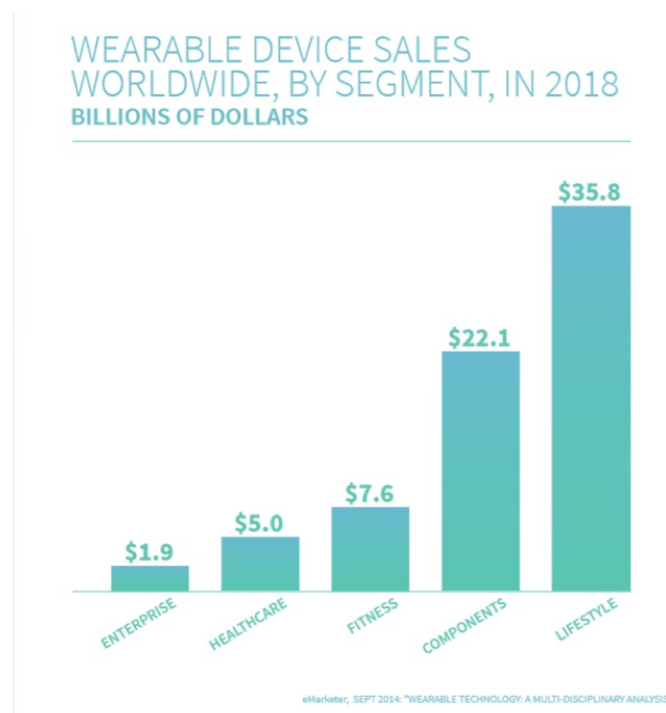


Figura 2.8: Vendas do Mercado *Wearable* mundialmente, por segmento, em 2018

a aparecer é a *Postura*, existem *wearables* (como o presente na figura 2.4) que tentam oferecer alertas aos utilizadores de quando estes estão sentados de forma incorreta ou com uma postura errada. Para além destas áreas já apresentadas, começa a aparecer *wearables* que tenta prever quando utilizadores (principalmente seniores) poderão cair, como por exemplo o *Kinesis QTUG* [19]. Até agora, todos contextos apresentados estão focados no bem-estar e a tentar melhorar o estilo de vida do utilizador, ou seja, não se focam a tentar combater uma doença específica, sendo mais difícil encontrar aparelhos deste tipo. Existindo por outro lado uma exceção à regra, que são os *wearables* que medem o nível de glucose, podendo oferecer métricas interessantes que ajudam utilizadores com diabetes. Sendo assim, juntamente com a *Glintt* foram identificados estes cinco contextos que representam uma força cada vez mais dominante e promissora no mercado atual, e que se enquadram no conceito de *Patient Generated Health Data* (conceito abordado em detalhe no capítulo 4.1):

- *Caminhada* – oferece métricas como batimento cardíaco, número de passos, escadas caminhadas, calorias queimadas, entre outras;
- *Sono* - métricas como a duração e consistência do sono ajudam o utilizador a perceber melhor a sua qualidade de sono;
- *Diabetes* – medir o nível de glucose sistematicamente e de uma forma simples (sem acesso ao sangue do utilizador), abre portas a um sistema de grande potencial;

- *Postura* – analisar a postura do utilizador permite avisá-lo de que não está a sentar-se ou a caminhar de forma correta;
- *Risco de queda* – oferece fortes métricas de quando o utilizador poderá cair devido a fatores como a fadiga, desequilíbrio, etc.

2.3.1 Caminhada

Juntamente com o mercado do *sono*, a *caminhada* é o mercado mais desenvolvido. Desde as *Fitbits* [14] até aos *Garmins* [15] é possível extrair métricas importantes como número de passos, calorias queimadas, batimento cardíaco, peso, escadas caminhadas e mais. Neste contexto as pulseiras inteligentes são o mais comum e mais acessível (de um ponto de vista monetário). Para além das pulseiras, também existem os *smartwatches* que oferecem um maior número de funcionalidades e tem um processamento computacional superior, mas que igualmente oferecem sensores para medir certos aspetos vitais dos utilizadores.

2.3.2 Sono

Grande parte dos *wearables* que oferecem métricas relacionados com a caminhada também avaliam a qualidade de sono do utilizador. Estágios de sono como acordado, REM³, leve e profundo (durante o tempo que o utilizador está a dormir) são alguns dos resultados que são apresentados aos utilizadores, como exemplo de um aparelho deste tipo existe o *wearable Fitbit Charge 2* [20].

2.3.3 Diabetes

Neste contexto o maior desafio é ler o nível de glucose sem utilizar uma amostra sanguínea. A maior parte dos aparelhos já consegue ultrapassar este obstáculo, surgindo alguns conceitos inovadores. Existem *wearables* que conseguem ler o nível de glucose através de um simples contacto com a pele [8]. Outro aparelho muito promissor são as *Google Contact Lenses* [21], que conseguem ler o nível de diabetes através das lágrimas, sendo que ainda não foram lançadas para o mercado.

2.3.4 Postura

Os aparelhos focados nesta área tentam avisar o utilizador quando este não está a caminhar ou sentado corretamente. Os avisos são feitos através de uma simples vibração ou através de notificações para o telemóvel. Outros *gadgets* deste tipo treinam a postura através de exercícios [22], funcionando com um *active coach* (treinador em tempo-real simulado através de um computador).

³Período de sono durante o qual pode ocorrer sonhos.

2.3.5 Risco de queda

Esta pode ser vista como a área menos desenvolvida e o principal motivo é porque todos os *wearables* funcionam como um sistema de emergência, simplesmente identificam quando o utilizador caiu. Ainda assim, existe um aparelho que consegue oferecer métricas muito interessantes chamado *Kinesis QTUG* [19] que funciona de forma diferente em comparação aos *wearables* deste contexto. *Kinesis QTUG* permite prever o momento que possivelmente o utilizador poderá cair, ou seja, oferece uma previsão.

2.4 Classificação Legal de Wearables nos EUA

Nesta secção é analisado como o processo de classificação de *wearables* é realizado nos Estados Unidos da América (EUA). Este tópico é importante, porque vão ser usados *wearables* como fonte principal de recolha de dados, para isso, o processo de classificação é um fator de seleção importante para a escolha dos *wearables*.

A organização que classifica os aparelhos médicos nos Estados Unidos da América é a *Food and Drug Administration* (FDA). FDA é uma agência federal do departamento de saúde e serviços Humanos dos EUA que está encarregue de proteger e promover a saúde pública. Engloba várias funções, sendo que uma delas é monitorizar os vários aparelhos médicos que podem ser comercializados. Antes de mostrar como o processo de classificação funciona (secção 2.4.2), é importante perceber o que é um aparelho médico e se todos os *wearables* podem ser considerados aparelhos médicos (secção 2.4.1).

2.4.1 Aparelho Médico

FDA classifica um aparelho médico como "um instrumento, aparelho, máquina, artifício, implante, reagente em *vitro*, ou outro artigo similar ou relacionado, incluindo uma parte de um componente ou acessório que é [23]:

- reconhecido oficialmente no *National Formulary*⁴, ou na *United States Pharmacopoeia*⁵, ou
- destinado a ser utilizado no diagnóstico de doença ou outras condições, ou na cura, mitigação, tratamento ou prevenção de doenças, no homem ou em animais, ou
- destinado a afetar a estrutura ou qualquer função do corpo do homem ou de animais, e que não alcança nenhum dos seus principais fins pretendidos através de ação química dentro ou sobre o corpo do homem ou de animais e que não depende de ser metabolizado para a realização de qualquer dos seus principais fins."

De seguida é abordado o debate entre FDA e *wearables*, mas antes é importante realçar que um aparelho médico "é usado no diagnóstico de doenças ou outras condições, ou na cura, mitigação,

⁴Entidade responsável por regular produtos farmacêuticos nos EUA.

⁵Outra entidade responsável por regular produtos farmacêuticos nos EUA.

tratamento ou prevenção de doenças, no homem ou em animais”. Sendo que é esta a diferença mais importante entre *wearables* e aparelhos médicos.

Para a FDA, um *wearable* para ser considerado um aparelho médico tem que provar que consegue tratar uma doença ou condição específica. Se não provar, é apenas considerado um *gadget* de bem-estar geral. Um *gadget* de bem-estar geral é um produto que promove um estilo de vida saudável, representando um risco muito baixo para a segurança dos utilizadores, por exemplo equipamentos de exercícios, gravações de áudio, videojogos, programas de software, entre outros produtos. [24]

2.4.2 Modelo de classificação

Antes de entrar no tópico de como o FDA classifica dispositivos médicos, é importante perceber alguns conceitos que serão usados no modelo de classificação:

- Classe I - é a classe que possui o menor risco associado para o utilizador e agrupa dispositivos médicos com controles gerais;
- Classe II - tem risco médio (moderado/controlado), os dispositivos desta classe também incluem controles gerais e controles especiais;
- Classe III - representa alto risco para o utilizador, também inclui controles gerais e especiais, mas a diferença para as outras classes é que estes dispositivos precisam de ser avaliados usando dados clínicos. Quase todos estes dispositivos precisam de passar por um processo chamado PMA (aprovação prévia ao mercado, descrito de seguida);
- Outros conceitos:
 - 510(k) - "A 510(k) é uma submissão de pré-mercado feita à FDA para demonstrar que o dispositivo a ser comercializado é seguro." [25];
 - Predicado - "Um aparelho que já é classificado pela FDA e serve com base de comparação para novos aparelhos que são introduzidos no mercado [25]";
 - De Novo 510(k) - "O processo De Novo fornece um modelo para classificar novos dispositivos médicos para os quais controles gerais, ou controles gerais e especiais, fornecem uma garantia razoável de segurança e eficácia para o uso pretendido, mas para o qual não existe um dispositivo de predicado legalmente comercializado. A classificação de De Novo é um processo de classificação baseado em risco. Os dispositivos que são classificados na classe I ou classe II através de um pedido de classificação De Novo podem ser comercializados e utilizados como predicados para futuras notificações de pré-notificação." [26];
 - Não-Predicado - "Um aparelho que não pode ser comparado a nenhum dos predicados. Consequentemente, é usado um processo diferente, chamado De novo 510(k), para classificar este novo aparelho. Se o processo tem sucesso, este aparelho passa a ser considerado um predicado." [26];

- HDU - "Um dispositivo médico destinado a beneficiar os pacientes no tratamento ou diagnóstico de uma doença/condição, que não afeta mais de 8000 indivíduos nos Estados Unidos por ano." [27];
- HDE - "Um pedido de comercialização para um HDU. Um HDE está sujeito a determinadas restrições de lucro e uso." [27];
- PMA - "A aprovação pré-comercial (PMA) é o processo da FDA de revisão científica e regulamentar para avaliar a segurança e eficácia dos dispositivos médicos de classe III." [28];
- Reclassificação - "O objetivo principal da reclassificação é aplicar o nível apropriado de controles de regulamentação para um tipo de dispositivo com base nas informações mais recentes sobre a sua segurança e eficácia." [29]

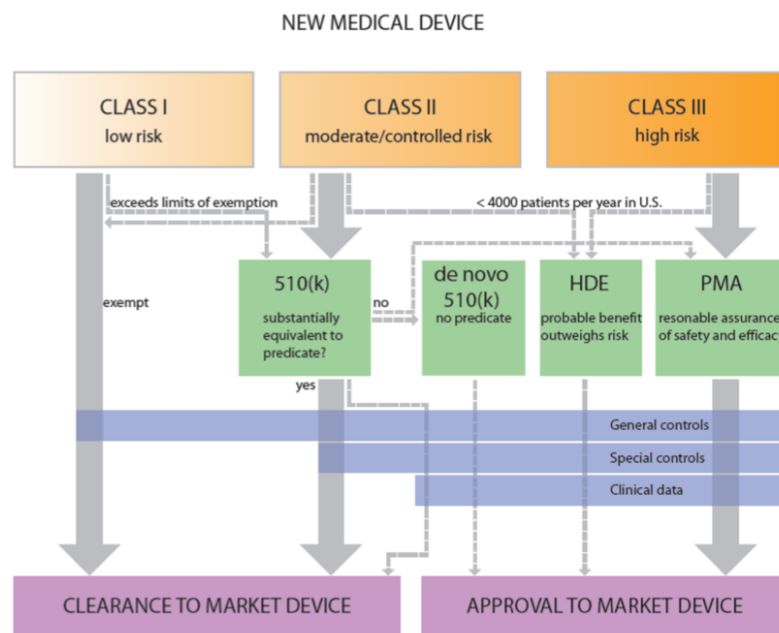


Figura 2.9: Modelo de classificação da FDA

De acordo com a FDA [30], um novo dispositivo precisa de ser introduzido em uma das três classes (classe I, II ou III). Se o dispositivo é de baixo risco para o utilizador e não excede os limites de isenção⁶, tem autorização para seguir para o mercado, se exceder os limites da isenção, passa por um 510(k) e, com sucesso, recebe a classe I. Se o dispositivo tiver um risco moderado/controlado e não exceder os limites de isenção tem autorização para o mercado, se exceder o limite da isenção, passa por um 510(k). Até agora o processo é igual ao da classe I, a diferença é que se o dispositivo for considerado um HDU, passa por um processo HDE. Relativamente a dispositivos de alto risco (classe III) existe dois caminhos para obter a aprovação ao mercado: HDE e PMA. Se o dispositivo for considerado uma HDU vai ao processo HDE, se não for passa por um PMA.

⁶Limite de isenção é um limite composto por vários critérios estabelecidos pela FDA.

2.5 Classificação Legal de *Wearables* na UE

Esta secção é semelhante à secção 2.4, mas está focado no processo de classificação de dispositivos médicos em território europeu, que inclui os seguintes países: Áustria, Itália, Bélgica, Letónia, Bulgária, Lituânia, Croácia, Luxemburgo, Chipre, Malta, República Checa, Países Baixos, Dinamarca, Polónia, Estónia, Portugal, Finlândia, Roménia, França, Eslováquia, Alemanha, Eslovénia, Grécia, Espanha, Hungria, Suécia, Irlanda e Reino Unido (em processo de saída da União Europeia).

A entidade, a nível europeu, que regula a comercialização de aparelhos médicos é a Comissão Europeia (CE) e para além de regular esta comercialização, também é responsável por propor a legislação, implementar decisões, manter os tratados da UE e gerir as operações de dia-a-dia.

2.5.1 Aparelhos Médicos

CE difere da FDA na definição de aparelho médicos, identificando quatro entidades:

- dispositivo médico;
- software;
- dispositivos médicos implantáveis ativos;
- dispositivos médicos em *vitro*;

O primeiro conceito é o de dispositivo médico, os dispositivos médicos são definidos como "qualquer instrumento, aparelho, material ou outro artigo, que é usado sozinho ou em combinação, incluindo o software necessário para a sua aplicação adequada, em seres humanos para vários fins, tais como diagnóstico, tratamento, alívio de doença, entre outros". Exemplos de dispositivos médicos incluem seringas, ligaduras, cadeiras de rodas, óculos de prescrição e solução de lentes de contacto, entre outros. Cobrem riscos mínimos, bem como riscos de elevado grau, a classificação do produto determinará se o fabricante pode auto-certificar ou precisa de passar um processo de laboratório de teste acreditado (caso falhe). [31] A CE também caracteriza o software, com um propósito médico, como um dispositivo médico por direito próprio. [31]

Os dispositivos médicos implantáveis ativos (AIMD), como os *pacemakers* cardíacos ou desfibriladores, são definidos como "qualquer dispositivo médico ativo que se destine a ser introduzido total ou parcialmente, cirurgicamente ou medicamente, no corpo humano ou por intervenção médica em um orifício natural, e que se destina a permanecer após o procedimento". Considerando o fator de potencial alto risco de estes dispositivos para o paciente, os fabricantes não se podem auto-certificar e têm que passar por um processo de laboratório de teste acreditado. [31]

Um dispositivo de diagnóstico em *vitro* (IVD) é um "reagente, produto reagente, calibrador, material de controle, kit, instrumento, aparelho, equipamento ou sistema, usado sozinho ou em combinação, destinado pelo fabricante a ser utilizado em *vitro*".⁷ Abrange itens como kits de teste de gravidez e máquinas de análise de sangue." [31]

⁷Em *vitro* significa que o produto é usado num ambiente controlado.

2.5.2 Modelo de classificação

O sistema de classificação retratado de seguida aplica-se apenas a dispositivos médicos. Em território europeu, os dispositivos podem ser introduzidos em quatro categorias: [31]

- Regras especiais (incluindo dispositivos médicos anticoncepcionais, desinfetantes e de diagnóstico radiológico).
- Dispositivos médicos invasivos.
- Dispositivos médicos ativos.
- Dispositivos não-invasivos.

Depois de introduzir o dispositivo em uma das quatro categorias, o dispositivo precisa de ser classificado. Existem cinco classes para esse fim: [31]

- Classe I:
 - Estéril ou não possui uma função de medição. Baixo risco;
 - Estéril e/ou tem uma função de medição. Risco baixo/médio.
- Classe *Ia* - Risco médio;
- Classe *Ib* - Risco médio/alto;
- Classe III - Risco alto.

Quando o aparelho é introduzido numa das secções é lhe atribuído um certificado que permite ser comercializado na União Europeia.

2.6 Sumário

Recapitulando, os *wearables* servem como fontes de recolha de dados para este projeto. Para abordar este tema, começou-se por definir o que caracteriza um aparelho deste tipo e foram apresentados resultados que o mercado destes aparelhos atingiu ao longo do tempo, servindo para justificar a sua relevância, (secções 2.1 e 2.2, respetivamente).

De seguida, foi explicado cada contexto em que os *wearables* devem atuar, sendo importante perceber que estas áreas foram escolhidas numa fase anterior ao desenvolvimento do projeto e foram identificadas pela *Glintt* com base no seu conhecimento adquirido ao longo dos anos (secção 2.3).

Por fim, foi feito o enquadramento deste tema com a área da saúde, através de como é efetuada a validação de aparelhos médicos, para que estes possam ser comercializados nos mercados americano e europeu. Foram escolhidos estes dois mercados por serem considerados os maiores para os quais se pretendem atuar. Ainda nestas duas secções (2.4 e 2.5), foram destacadas as principais

diferenças entre aparelhos médicos e *wearables*, para se perceber se todos os *wearables* devem passar por estes processos de validação (secção 2.4.1).

O próximo capítulo (3) irá abordar aspetos da segurança de informação e regulamentações legais que possam afetar a solução desenvolvida, devido ao facto de os dados usados serem dados de natureza sensível, ou seja, dados relativos à saúde de indivíduos.

Capítulo 3

Dados de saúde

Os dados recolhidos pelos *wearables* são considerados dados sensíveis, devido à sua natureza (dados de saúde), fazendo com que seja peremptório abordar aspetos relacionados com a segurança da informação. Sendo assim, esta secção está segmentada da seguinte maneira:

- na secção 3.1 é abordado um conceito genérico de segurança da informação (Tríade CIA);
- a secção 3.2 foca-se em dois *standards* que formalizam diretrizes globais, que as organizações devem usar para atingir segurança da informação nos seus sistemas;
- na penúltima secção (3.3) é abordado um tema muito recente (tendo entrado em vigor na reta final deste trabalho) que vai mudar a forma como os dados pessoais devem ser tratados e transmitidos entre controladores (entidades que manipulam e trabalham informação de pessoas ou entidades) na zona da união europeia;
- a última secção (3.4) tem como objetivo focar e relembrar os temas principais deste capítulo.

3.1 Tríade CIA

Tríade CIA serve como modelo conceptual para sumarizar três aspetos importantes da segurança computacional e de informação [32], que são os seguintes:

- Confidencialidade.
- Integridade.
- Disponibilidade (em inglês, *Availability*).

3.1.1 Confidencialidade

Confidencialidade define que os utilizadores que tem permissão para aceder a uma certa informação são os únicos que efetivamente acedem à informação. Basicamente, consiste na proteção da informação a acessos não autorizados, requerendo medidas de segurança para garantir que apenas

as pessoas autorizadas tenham acesso a ativos sensíveis. Tecnicamente, confidencialidade pode ser conseguida através de [33]:

- Protocolos seguros;
- Encriptação de dados críticos;
- Autorização;
- Implementação de um programa detalhado de gestão de vulnerabilidades para identificar e capturar falhas (para cobrir vulnerabilidades em implantações de *patches*, por exemplo).

3.1.2 Integridade

Integridade é uma condição que está relacionada com alterações não-autorizadas de dados, e onde a informação é mantida precisa e consistente, a não ser que alterações autorizadas sejam efetuadas. Quando a integridade é danificada, algumas das razões podem ser: o acesso e uso descuidado, erros no sistema ou acesso e uso não autorizado. Através da segurança da integridade é possível [34]:

- Impedir utilizadores não autorizados de fazer modificação de dados;
- Prever utilizadores não autorizados de fazer alterações impróprias à informação do sistema;
- Manter a consistência interna ou externa dos dados.

3.1.3 Disponibilidade, em inglês *Availability*

Disponibilidade pode ser medida como o tempo em que o sistema precisa para responder aos utilizadores e é mantida quando todos os componentes do sistema trabalham de forma correta. Sendo que assenta num pilar importante: a informação deve estar sempre disponível quando utilizadores autorizados lhe precisam de aceder. A disponibilidade do sistema lida com vários problemas, como por exemplo: [34]

- Negação de serviço (DoS) devido a ataques intencionais ou a falhas de implementação;
- Perda de capacidades do sistema por causa de desastres ambientais (incêndios, inundações, tempestades ou terremotos) ou ações humanas (por exemplo bombas);
- Falhas nos equipamentos durante o uso.

3.2 Standards de segurança

Esta secção detalha dois *standards* da segurança dedicados à segurança da informação de organizações. O primeiro (ISO/IEC 27002:2103) trabalha com informação sensível de qualquer

domínio e o outro *standard* (ISO 27799:2016) é especializado a tratar dados de saúde. Ambos os *standards* tem uma estrutura similar, sendo que ISO 27799:2016 é derivado do ISO/IEC 27002:2013.

3.2.1 ISO/IEC 27002:2013

O ISO/IEC 27002:2013 fornece uma estratégia para gerir a informação organizacional e práticas de segurança, envolvendo a seleção, implementação e gestão de controlos tendo em conta a informação da organização num ambiente de risco. [35] Este documento foi escrito para ser usado em organizações que planeiam:

- escolher controladores no processo de implementação de um sistema de gestão de informação segura;
- implementar controladores da segurança da informação;
- cultivar procedimentos próprios de gestão de segurança da informação.

Permite também:

- identificar requisitos da segurança da informação;
- seleccionar controladores;
- desenvolver diretrizes próprias;
- considerar o ciclo de vida da informação.

Identificar requisitos de informação segura depende do contexto da organização. Sendo assim, existe três fontes de requisitos a ter em conta:

- Avaliação dos riscos para a organização, levando em consideração a estratégia e os objetivos globais da organização;
- Requisitos legais, regulamentares e contratuais a que uma organização precisa de obedecer;
- Conjunto de princípios, objetivos e requisitos de negócios para tratamento, processamento, armazenamento e comunicação da informação.

A seleção de controladores pode ser feita através deste documento ou de novos controladores que podem ser considerados. Este *standard* tem de ser considerado um ponto de partida para que as organizações possam desenvolver as suas próprias diretrizes, para melhorar processos existentes.

O último aspeto a considerar é o ciclo de vida da informação. A informação tem um ciclo de vida natural, desde a criação e origem até ao armazenamento, processamento, uso e transmissão e até a uma eventual destruição ou declínio. Portanto, é importante perceber que os recursos podem variar durante a sua existência, por isso o ciclo de vida da informação é um aspeto importante a ser considerado.

3.2.2 ISO 27799:2016

O ISO 27799:2016 direciona a implementação de processos para os controladores descritos no ISO/IEC 27002 e melhora-os quando necessário, para que possam, efetivamente, ser usados para manipular informação de saúde de uma forma segura. Implementando o ISO 27799:2016, as organizações de cuidados de saúde e outros detentores de informações de saúde serão capazes de garantir um nível mínimo de segurança que seja correto para as circunstâncias da organização e manterão a confidencialidade, integridade e disponibilidade (tríade da CIA) de informação de saúde pessoal. Aplica-se a qualquer tipo de informação de saúde (desde palavras a números, incluindo gravações de som, vídeo e outros) quando a informação está a ser armazenada e transmitida. [36]

3.3 Regulação Geral da Proteção de Dados (RGPD)

O RGPD regula a proteção de dados eletrónicos em território europeu, sendo que não vai apenas regular processos dentro da UE, mas também vai cobrir transações fora da UE que possam afetar a privacidade de cidadãos da UE [37]. Esta secção está dividida em outras duas secções:

- A primeira (3.3.1) explica quais os dados que devem ser protegidos.
- O outra (3.3.2) descreve como os dados devem ser transmitidos entre controladores de informação.

De seguida, são apresentados vários factos que justificam como o RGPD é importante e perigoso (em caso de falha) para as empresas [38]:

- Multas de 4% da receita anual da empresa mãe;
- O RGPD vai afetar todas as empresas de tratamento de dados pessoais de utilizadores que vivem na EU, independentemente da localização da empresa;
- O utilizador deve poder remover o consentimento de partilha ou manipulação dos seus dados de uma forma fácil e simples;
- O controlador deve permitir ao utilizador apagar os seus dados pessoais do sistema, assim como terminar a distribuição dos mesmos;
- Falhas que resultem em risco para os direitos e liberdade dos indivíduos, devem ser reportadas dentro de 72 horas depois do administrador do sistema se aperceber de tal falhas.

3.3.1 Tratamento de dados de forma segura

Existe várias técnicas que permitem o tratamento e segurança dos dados, mas o mais importante é perceber quando e quais são os dados que precisam de ser seguros. O Artigo 35 [39] do RGPD e a diretiva 2016/680 [40] introduz o conceito de Avaliação de Impacto de Proteção de

Dados (em inglês, *Data Protection Impact Assessment*, de acrónimo DPIA). DPIA é um procedimento projetado para medir a necessidade e proporcionalidade de um processo, permitindo assim ajudar a gerir riscos aos direitos das pessoas, resultante do processamento de dados pessoais. Em outras palavras transcritas do Artigo 35(7), DPIA é considerado [39]:

- "Uma descrição sistemática das operações previstas para o processamento, incluindo, quando aplicável, o interesse legítimo do controlador;
- uma avaliação da necessidade e proporcionalidade das operações de processamento;
- uma avaliação dos riscos para os direitos e liberdades dos sujeitos em causa;
- as medidas previstas para enfrentar os riscos, incluindo salvaguardas, medidas de segurança e mecanismos para garantir a proteção dos dados pessoais e demonstrar a conformidade com o regulamento, tendo em conta os direitos e os interesses legítimos das pessoas em causa e outras pessoas que possam ser afetadas";

Com a informação anterior, é possível perceber que o DPIA ajuda as organizações a determinar quando e quais são os procedimentos que podem afetar os direitos dos utilizadores descritos no RGPD [39]. Portanto, tenta ajudar a perceber qual é a informação sensível que deve ser tratada de uma forma segura, para proteger os direitos dos utilizadores a futuras violações do RGPD.

Esta secção está estruturada em três secções: a primeira define o conceito DPIA (3.3.1.1), a segunda (3.3.1.2) descreve quais são os processos obrigados a aplicar o DPIA e, finalmente, explica como implementar o DPIA (3.3.1.3).

3.3.1.1 DPIA explicado

De acordo com a regulamentação 2016/679 [41] (figura 3.1), a primeira questão que se levanta é a seguinte: o processo em questão "é provável de resultar em riscos altos" para os direitos e privacidade da informação do utilizador? Se não, o DPIA não é necessário. Mas se a resposta for sim, é importante perceber se existe uma exceção à regra, se não existir, então o DPIA tem de ser implementado.

O DPIA aplica-se a um processo singular de operação de dados, embora possa ser usado em várias operações que são similares em termos de riscos. Terá de ter sempre em conta a natureza, âmbito, contexto e os objetivos desse processo. Quando o processamento inclui vários controladores é importante definir as obrigações de cada um de uma forma muito precisa. No DPIA deve-se definir qual controlador que é responsável pelas várias medidas destinadas a tratar os riscos e proteger os direitos das pessoas em causa.

3.3.1.2 Processos exemplificativos sujeitos ao DPIA

O DPIA, como o artigo 35(1) [39] explica, é apenas obrigatório em processos que "provavelmente resultem em riscos altos para os direitos dos pessoas". Por outro lado, quando não está claro

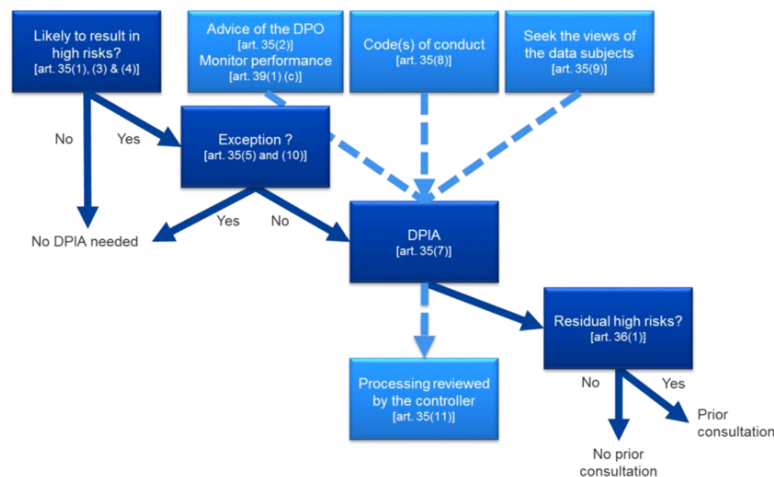


Figura 3.1: DPIA explicado

se o DPIA deve ser implementado, opta-se pela implementação, porque ajuda os controladores a estarem de acordo com a lei.

Às vezes, é difícil de entender quando é necessário implementar o DPIA, para isso seguem-se alguns exemplos:

- Avaliação ou classificação (como por exemplo, entidades bancárias que comparam dados dos seus clientes a uma base de dados de cartões de créditos já existente);
- Tomada de decisão automatizada com efeito legal ou similar significativo (processo que pode levar à exclusão ou discriminação contra alguns utilizadores);
- Monitorização sistemática (acontece quando os dados pessoais são colecionados em circunstâncias onde os utilizadores não estão conscientes de quais dados são colecionados e para que vão ser usados);
- Dados sensíveis (como por exemplo dados de saúde, opiniões políticas ou dados criminais);
- Dados processados em larga escala (larga escala é considerada quando o número de sujeitos afetados é uma porção relevante da população);
- Conjunto de dados que são combinados (por exemplo, dados originados por duas ou mais operações de processamento para diferentes propósitos e/ou por diferentes controladores de dados de forma a exceder as expectativas razoáveis da pessoa em causa);
- Dados relativos a indivíduos vulneráveis (por exemplo, funcionários que costumam encontrar dificuldades fortes para resistir ao processamento executado pelo seu empregador. Além disso, as crianças podem ser consideradas incapazes de se manifestar conscientemente ao processamento dos seus dados);

- Uso inovador de soluções tecnológicas e organizacionais (por exemplo, impressão digital e reconhecimento facial);
- Transferências de dados para além da fronteira da União Europeia.

Antes de entrar na próxima secção, é importante de perceber que cada processo que, como o Artigo 22 [39] explica, "impeça que os utilizadores exerçam o seus direitos ou (que usem um serviço) através de um contrato" são condenáveis pela lei.

3.3.1.3 Como implementar o DPIA

O DPIA deve ser implementado "antes do processamento" de acordo com os artigos 35(1) e 35(10) [39], recitais 90 e 93. O DPIA tem que ser implementado o mais cedo possível antes e depois do desenho do processo, também deve ser atualizado durante o ciclo de vida do projeto. Às vezes, é necessário repetir as etapas individuais da avaliação à medida que o processo de desenvolvimento avança, devido ao facto de a seleção de certas decisões poderem afetar a gravidade ou a probabilidade de riscos. Aplicar o DPIA é um processo recorrente e não uma aplicação única, como mostra a figura 3.2 [41].

O RGPD cria um conjunto de características mínimas em que um DPIA deve consistir, como o Artigo 35(7) nos recitais 84 e 90 demonstra [39]:

- "uma descrição das operações de processamento previstas e os objetivos do processamento;
- avaliação da necessidade e proporcionalidade do processamento;
- avaliação dos riscos para os direitos e liberdades dos utilizadores em causa;
- as medidas previstas para abordar os riscos em conformidade com a regulamentação”.

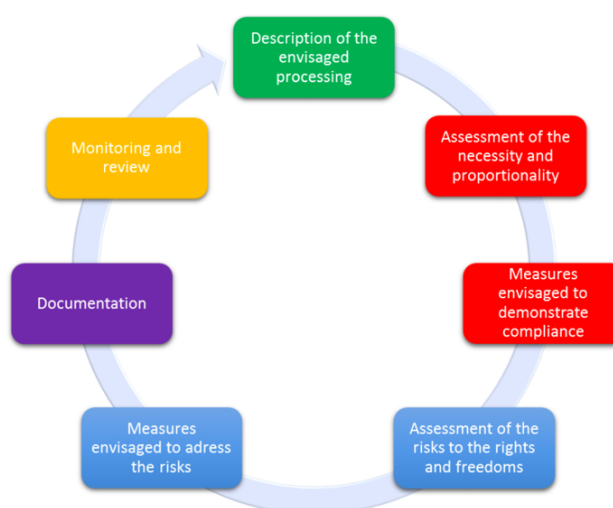


Figura 3.2: Processo de DPIA

3.3.2 Transmissão dos dados de uma forma segura

Para além de definir quando e que informação e direitos devem ser protegidos, RGPD também explica como a portabilidade de dados funciona. Em outras palavras, como a transmissão de informação entre controladores de dados deve funcionar.

Primeiramente, é definido o que é portabilidade de dados (3.3.2.2) e quando se aplica e, por fim como os dados devem ser fornecidos de um controlador para outro (3.3.2.1).

3.3.2.1 Definição de portabilidade de dados

O Artigo 20 [39] do RGPD define que a portabilidade de dados como "a pessoa em causa deve ter o direito de receber os dados pessoais referentes a ele/ela num formato estruturado, legível por uma máquina. Tendo o direito de transmitir esses dados para outro controlador sem nenhum obstáculo do controlador ao qual os dados foram fornecidos".

Portabilidade de dados tem as seguintes características:

- O direito de receber dados pessoais;
- O direito de transmitir dados pessoais de um controlador para outro;
- O direito de controladoria.

Qualquer sujeito de dados (utilizador) deve ter a capacidade de armazenar suas informações para uso pessoal, o armazenamento pode ser num dispositivo privado ou numa *cloud* privada, sem a necessidade de transmitir os dados entre controladores. Por exemplo, um utilizador tem o direito a exportar os seus dados pessoais e a aplicação onde a informação está a ser exportada não necessita de transmitir diretamente para outra aplicação, só precisa de simplesmente fornecer serviços ao utilizador para que ele/ela possa exportar os seus dados.

Por outro lado, um utilizador tem de ter a capacidade de transmitir os seus dados pessoais de um controlador para outro, sem obstáculos desnecessários. Com base no pressuposto de que é tecnicamente viável, um controlador de dados deve transmitir diretamente as informações para outros controladores, a pedido do utilizador.

Em termos de controladoria, deve-se definir em que lado a responsabilidade deve estar, o Artigo 20 [39] define que os controladores que respondem a pedidos de portabilidade de dados não são responsáveis pelo processamento dos dados do utilizador em causa ou por outra empresa que recebe esses dados pessoais (se agirem em nome do utilizador em causa). Numa situação em que os dados pessoais solicitados são tratados por um processador de dados, o controlador tem a responsabilidade de auxiliar o "outro controlador em todas as questões técnicas e organizacionais necessárias, (...) para responder aos pedidos de exercício dos direitos do utilizador em causa"(de acordo com o Artigo 28 [39] do RGPD). O controlador deve, conseqüentemente, implementar procedimentos explícitos de assistência para suportar os seus processadores de dados a responder a futura solicitações.

3.3.2.2 Quando aplicar?

Baseado no Artigo 20 [39] do RGPD, portabilidade dos dados deve ser aplicado em duas situações:

- no consentimento da pessoa em causa;
- num contrato no qual a pessoa em causa faz parte.

Quando nenhuma das condições são verificadas, não há obrigação para o controlador permitir a portabilidade de dados. Por exemplo, não há compromisso para um banco atender um pedido de portabilidade de dados para detetar o branqueamento de capitais.

Outro aspeto importante é compreender quais as informações que devem ser transmitidas, que incluem dados relativos ao utilizador e dados fornecidos pelo utilizador. Não devem ser considerados casos em que a informação seja anónima (fonte desconhecida). Os dados fornecidos pelo utilizador englobem nomes de utilizadores, idade, endereço de e-mail e muito mais. Também inclui dados pessoais que são considerados "observados", como histórico de pesquisas, dados de tráfego e dados de localização.

3.3.2.3 Como fornecer dados portáveis?

RGPD [39] define que os dados devem ser fornecidos sem obstáculos desnecessários, um obstáculo inclui qualquer entrave legal, técnico ou financeiro colocado pelo controlador para abster ou diminuir o acesso, transmissão ou reutilização, por exemplo falta de interoperabilidade, atraso excessivo e complexidade. Também define que a transmissão deve ser considerada quando é "tecnicamente viável" e cada caso deve ser tratado individualmente. Recital 68 [39] clarifica que "não se deve criar uma obrigação para os controladores adotarem ou manterem um sistema que seja tecnicamente compatível em todas as frentes".

Obviamente, um controlador não precisa de exportar dados para cada controlador existente, se a conexão direta for possível deve acontecer. Quando não é possível, a exportação pode ser feita por uma ferramenta automatizada que permite a extração de dados em formatos padrão como CSV, XML, JSON e muito mais.

De acordo com RGPD [39], o formato deve ser "legível por máquina", considerando o recital 21 da diretiva 2013/37/EU [42] define "legível por máquina" como "um formato de arquivo estruturado para que as aplicações de software possam facilmente identificar, reconhecer e extrair dados específicos, incluindo declarações individuais de factos e estruturas internas. Os dados codificados em arquivos estruturados e num formato legível por máquina são dados legíveis por máquina. Os formatos legíveis por máquina podem ser abertos ou proprietários. Documentos codificados em um formato de ficheiro que limitam o processamento automático, porque os dados não podem ser extraídos ou não podem ser facilmente obtidos, não devem ser considerados em um formato legível por máquina."

É importante entender que a portabilidade tenta produzir sistemas interoperáveis, não sistemas compatíveis, portanto os sistemas devem poder comunicar entre si para extrair apenas a informação desejada. Dito isto e considerando o artigo 68, interoperável é definido como "a capacidade de diferentes organizações de interagirem para atingir objetivos comuns mutuamente benéficos e acordados, envolvendo o compartilhamento de informações e conhecimento entre as organizações".

O último ponto a ser abordado é como a segurança de dados portáteis deve ser tratada. Quando a comunicação é feita através de um controlador para o outro, o controlador de dados a partir do qual as informações são fornecidas é responsável por tomar todas as medidas de segurança necessárias. Não deve garantir apenas que os dados sejam transmitidos de forma segura e para o destino certo, mas também deve garantir a proteção os dados que permanecem nos seus sistemas, bem como procedimentos transparentes para lidar com possíveis violações de dados. Quando a informação é transmitida para outro controlador, o controlador recetor deve proteger a informação quando está presente no seu lado, assim a informação fica duplicada e segura em ambos os lados. Quando a informação é exportada pelo armazenamento através da ação do utilizador, a pessoa em causa é responsável por reconhecer as técnicas certas para garantir a segurança dos seus dados. No entanto, o utilizador deve estar consciente de tal, para tomar essas medidas (por exemplo, ferramentas de criptografia e outros procedimentos de segurança para ajudar o utilizador) para garantir a segurança da informação.

3.4 Sumário

Este capítulo tem como objetivo enquadrar o leitor na perspetiva da segurança da informação e como esta pode ser obtida. Portanto, é feita uma introdução a um conceito que serve para perceber as 3 principais características base da segurança de informação (3.1). De seguida, foram abordados dois documentos que agregam diretrizes globais para atingir este tipo de segurança (3.2), tanto a nível geral como, especificamente, para dados de saúde. Na última secção (3.3), foi abordada a nova regulamentação europeia de dados pessoais (que inclui dados de saúde, obrigatoriamente). Através da leitura deste capítulo, é possível entender vários tópicos considerados marcantes a nível da segurança e da legislação. No próximo capítulo (4), é explicado o tema central deste projeto e também é contextualizado a par com o conceito dos *wearables*.

Capítulo 4

Patient Generated Health Data com Wearables

Neste capítulo vai ser enunciado o problema que esta dissertação se propõe a resolver, com esse objetivo em mente o capítulo está dividido em quatro secções:

- 4.1 define de uma forma clara e objetiva o problema desta dissertação, *Patient Generated Health Data (PGHD)*;
- 4.2 aborda os vários desafios que este problema impõe, explicando a ligação deste tema com o conceito dos *wearables* (já abordado na secção 2);
- 4.3 define que tipo de soluções este problema já tem no mercado;
- 4.4 explica em que medida a solução pretendida sobressai em relação às existentes;
- 4.5 resume o capítulo, fazendo também a ponte para os capítulos seguintes (5, 6, 7 e 8).

4.1 *Patient Generated Health Data (PGHD)*

Até agora foi abordado um tema central deste projeto (capítulo 2) que são os *wearables*, estes aparelhos funcionam como poderosas fontes de recolha de dados. Mas por si só não constituem o tema principal do projeto, falta ainda perceber o aspeto diferenciador desta dissertação.

O objetivo principal deste projeto é desenvolver uma aplicação (que suporta duas plataformas: *iOs* e *Android*) na qual um utilizador tem acesso a um conjunto de funcionalidades. A aplicação vai utilizar APIs que consomem dados dos *wearables* que, por sua vez, estão inseridas num (ou em mais do que um) dos cinco contextos já enunciados na secção 2.3. O aspeto diferenciador é o âmbito em que esta a aplicação vai ser utilizada. A maior parte dos utilizadores adquirem o *wearable* e quase a totalidade dos fabricantes fornecem uma aplicação, que os utilizadores instalam e usam para poderem visualizar os dados. O que é pretendido fazer neste projeto é estudar e perceber o uso deste tipo de aplicações num contexto clínico sem que o paciente esteja lá presente.

De uma forma simples, é pretendido oferecer aos agentes de saúde uma forma de monitorizar os pacientes sem que estes estejam num ambiente clínico, nascendo um conceito conhecido como *Patient Generated Health Data* (ou simplesmente PGHD) ou em português "Dados de saúde gerados pelo paciente". O objetivo é perceber o potencial deste crescente conjunto de dados que milhares de pacientes geram e como podem servir os agentes de saúde. Com isto, levantam-se uma série de questões legais e técnicas que têm de ser tratadas. Para se ficar a perceber melhor este conceito, esta secção divide-se em três partes: a primeira define PGHD (4.1.1), a segunda (4.1.2) aborda as vantagens e, por fim, (4.1.3) os desafios que este tema tem por vista ultrapassar.

4.1.1 Definição

Segundo a ONC (*Office of the National Coordinator for Health Information Technology*) [43], PGHD é definido como "dados relacionados com a saúde dos pacientes criados, registados ou obtidos por pacientes (ou membros da família) para ajudar a tratar problemas de saúde. Inclui histórico de saúde e tratamentos, dados biométricos, sintomas e escolhas relacionados com o estilo de vida do paciente". Também ainda é referido quais são fatores que distinguem este tipo de dados com dados obtidos em ambiente clínicos:

- Os próprios pacientes são os principais responsáveis por capturar os dados.
- Os pacientes decidem como partilhar ou distribuir os dados para agentes de saúde.

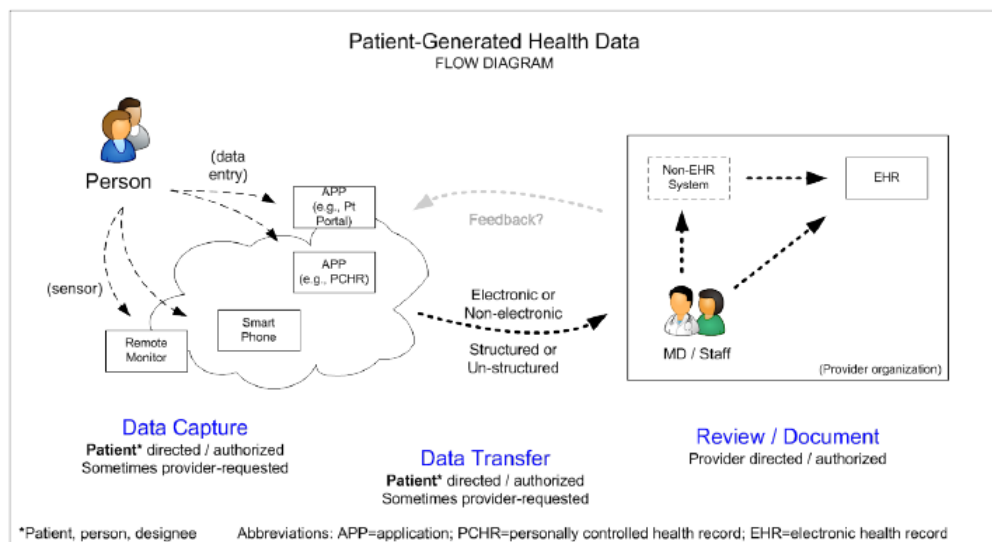


Figura 4.1: PGHD fluxo de dados

Como a figura 4.1 [44] mostra, este processo pode ser dividido em três tarefas: captura dos dados, transferência dos dados e revisão/documentação.

O processo de captura inclui toda a criação e armazenamento dos dados de saúde por intermédio do paciente, os dados podem ser introduzidos por escrita (através do teclado), verbalmente

(microfone) ou mesmo dados fisiológicos e/ou ambientais obtidos por via de um dispositivo de monitorização (como por exemplo, um *wearable*).

A transferência inclui todo o processo de comunicação dos dados desde o paciente (fonte) até aos agentes de saúde, esta transferência pode ser feita eletronicamente (via e-mail ou por outro meio através da internet), por telefone ou por reunião presencial com o médico.

A revisão dos dados é um processo que foge ao âmbito deste projeto, porque acontece quando o agente de saúde revê os dados e decide o que fazer com eles. Sendo assim, nesta dissertação a tarefa que se pretende automatizar/facilitar é a de transferência (sendo que a recolha já é feita pelos *wearables*).

Existem vários contextos em que PGHD atua, como por exemplo:

- Paciente (ou um familiar do mesmo) mede os dados vitais através de um aparelho. Os dados podem ser temperatura, pressão sanguínea, nível de glucose, peso, etc.
- Dados relacionados com estilo de vida recolhidos pelo paciente. Os dados são recolhidos manualmente, podem ser dados de ingestão calórica, dieta, exercício, hidratação, adesão à medicação, entre outros.
- Dados relacionados com a qualidade de vida recolhidos pelo paciente ou por um familiar. Exemplos deste tipo de dados pode ser humor, qualidade de sono, nível de dor, contactos sociais, entre outros.

Apesar do foco crescente dado atualmente a este tema, já é um conceito com alguns anos de existência, mesmo que antigamente não fosse reconhecido por este nome. Por exemplo, um médico sentiu a necessidade de monitorizar o paciente num dia inteiro, então o médico pediu ao paciente para levar para casa um aparelho que media o nível de diabetes. Por sua vez, o paciente levava o aparelho e num bloco de notas anotava as leituras.

Neste projeto o que se pretende obter é: desenvolver um sistema que consegue automatizar o processo de transferência dos dados que chegam até ao agente de saúde. Assim, é possível corrigir eventuais erros que o paciente possa cometer, sem que este necessite de levar até ao agente de saúde os dados recolhidos na próxima visita.

4.1.2 Vantagens

Existem várias vantagens relativas a este conceito, na perspetiva do paciente temos as seguintes:

- consegue ter o processo de recolha de dados facilitado;
- se já possuir o aparelho (como por exemplo um *smartwatch* ou uma pulseira inteligente) não necessita de configura-lo;
- não necessita de entregar nada ao agente de saúde no fim do período da recolha;

- melhora a saúde do paciente, através da entrega de uma monitorização contínua de dados, abrindo portas a outras interpretações por parte do agente de saúde;
- o número de visitas ao agente de saúde podem ser menores.

O outro tipo de ator envolvido neste processo são os agentes de saúde, as vantagens que o PGHD trás para os agentes de saúde são:

- oferece um conjunto maior de dados que permite monitorizar melhor a saúde do utente;
- se o aparelho for certificado por entidades reguladoras, garante maior qualidade e precisão dos dados recolhidos;
- se o utente já possuir o aparelho, o agente de saúde não necessita de o fornecer, bem como não perde tempo a ensinar o paciente a configura-lo;
- é sempre melhor que os dados sejam recolhidos por um aparelho certificado do que pelo paciente, para prevenir erros do lado do utente na recolha dos dados;
- um estudo feito em Arizona revela o PGHD pode reduzir o número de visitas a hospitais cerca de 64% e pode reduzir os custos por 72%. [45].

4.1.3 Obstáculos

Por outro lado, PGHD introduz algumas barreiras culturais ou mesmo técnicas. O facto de os dados estarem sempre disponíveis e em tempo-real para o médico, faz com que crie uma responsabilidade acrescida para o agente de saúde de estar constantemente a rever os dados do paciente. O processo de revisão dos dados não pode ser automatizado, porque a última palavra deve caber sempre ao agente de saúde. De um ponto de vista técnico, podem existir outros entraves como [46]:

- incorporação deste sistema em outros já existentes;
- questões de legais que podem surgir por se tratar de dados de saúde.

Certos pacientes podem não ter conhecimentos suficientes para usar o telemóvel (por exemplo idosos, pessoas com limitações físicas). O facto de os dados serem tratados antes pelos fornecedores dos *wearables* podem levantar questões relativamente à privacidade dos utentes, fazendo com que alguns destes fornecedores tenham um conjunto de responsabilidades acrescidas, porque os dados recolhidos não vão ser observados por utilizadores normais, mas sim por agentes de saúde. Uma preocupação grande é o facto de alguns dos agentes de saúde estarem reticentes relativamente à precisão e exatidão dos aparelhos, justificando que alguns aparelhos não recolhem os dados de forma totalmente correta. [45]

4.2 Desafios

Essencialmente, o problema passa por criar um sistema que permita agilizar o processo de transferência de dados de saúde e bem-estar do paciente, sem que estes dados sejam gerados num ambiente clínico e em contacto direto com um agente de saúde. Ou seja, é necessário criar um sistema que permita colmatar o fosso de comunicação entre pacientes e agentes de saúde para além das paredes de um hospital ou clínica.

Por outras palavras, é necessário criar uma prova de conceito¹ que permita explorar o potencial dos dados criados por utilizadores (que nos últimos anos tem vindo a ser acentuado devido ao uso dos *wearables*) sem que estes estejam na presença de um agente de saúde.

Neste projeto existem dois tipos de atores: o paciente e o agente de saúde (médico, nutricionista, fisioterapeuta, entre outros). Sendo que do ponto de vista tecnológico será desenvolvido uma aplicação para servir apenas o paciente. Numa segunda fase (que não faz parte do âmbito desta dissertação), será incorporado o papel do agente de saúde. Mas é importante perceber, que numa primeira fase o sistema tem que ser desenhado e implementado de modo a que permita fáceis modificações futuras.

Como já referido, PGHD está a ganhar cada vez mais importância e não é por acaso que esta evolução vem a ser mais notória na mesma época em que os *wearables* têm vindo a evoluir cada vez mais. Com o grande interesse nos *wearables* e com um crescente conjunto de dados a ser gerado dia após dia, começa-se a perceber que existe muita informação com um grande potencial a ser pouco explorada. Informação esta, que se encaixa na perfeição no conceito de PGHD, chegando-se á conclusão de que o processo de recolha já está a ser feito.

Para concluir, o objetivo é aproveitar a informação já gerada por um conjunto grande de utilizadores e recolhida pelos *wearables*, e provar que é possível visualizar essa informação numa aplicação móvel, sendo que essa informação é gerada por várias plataformas de fontes de dados de diferentes fornecedores, que tratam a informação de forma distinta.

4.3 Trabalho relacionado

Existe um exemplo prático que consegue introduzir o conceito PGHD num ambiente clínico já existente: *FormFast Connect* (uma tecnologia de *eForms*²) para os hospitais. Sendo assim, com o *FormFast Connect* é possível incorporar a informação dos *eForms* diretamente no sistema EHR do hospital, ficando visível para a equipa constituída pelos agentes de saúde. A informação do *eForm* é preenchida pelo utilizador e contempla informação de saúde relativa ao mesmo. Esta informação introduzida pelo paciente no *eForm* enquadra-se no conceito de PGHD, devido ao facto de ser informação de saúde e bem-estar do paciente que não é gerada num ambiente clínico. E através da tecnologia *FormFast Connect* é introduzida diretamente no sistema informático do hospital ou clínica, ou seja, *FormFast Connect* permite automatizar o conceito de PGHD. [47]

¹ Prova de conceito é um conceito utilizado para descrever um modelo prático que prova um conceito teórico através de uma pesquisa.

² Formulário em formato eletrónico.

A *Apple* [48] também desenvolveu um sistema que consegue ficar perto do que é pretendido neste projeto, que tem como nome: *Apple Healthkit* [49]. Basicamente, *Healthkit* é uma *framework*³ desenvolvida pela *Apple* que permite desenvolver aplicações de *fitness* e saúde. Ou seja, permite que contribuidores desta plataforma desenvolvam aplicações usando os dados de saúde de utilizadores que são recolhidos por *wearables* da *Apple*, como por exemplo *Apple Watches*. Um exemplo prático de uma aplicação desse tipo é a *Epic Haiku* [50]. *Haiku* permite que utilizadores de clínicas com sistema EHR⁴ da *Epic* tenham acesso seguro a marcações, lista de pacientes, resultados de testes, notas, entre outros. Para além do *Healthkit*, a *Apple* também fornece outros serviços de saúde como *ResearchKit* [51], que através de dados de saúde dos utilizadores, permite fornecer uma grande base de dados para médicos usarem para pesquisa.

Para fechar esta secção, é importante referir outro produto denominado por *S-Patch* que ainda não foi lançado para o mercado, mas que já foi anunciado pela *Samsung*⁵ [18] numa conferência chamada *Consumer Electronics Show*⁶ que decorreu em 2016. Este aparelho está constantemente a monitorizar o batimento cardíaco do utilizador, transmitindo essa informação para o telemóvel do utilizador, ou até mesmo para um médico. Sendo assim, consegue aproveitar informação clínica não gerada num ambiente clínico que pode ser avaliada por um agente de saúde [52]. Com isto é possível perceber que este conceito é, em comparação com os outros, o que consegue chegar mais perto ao conceito de PGHD.

4.4 Fator diferenciador da solução

Como foi descrito na secção 4.3, existem implementações do conceito PGHD, como o *FormFast Connect*. O que diferencia esta solução do *FormFast Connect* é que para o processo de recolha de dados vão ser utilizados *wearables*, que conseguem recolher os dados com o mínimo esforço do utilizador.

As outras soluções tratadas na secção 4.3 que atacam este problema são *Apple HealthKit* e *S-Patch*. O fator que distingue a solução pretendida neste projeto da fornecida pela *Apple* [48], é o facto de trabalhar com *wearables* de diferentes fornecedores. Devido ao facto de existir um grande número de utilizadores que não utilizam tecnologia da *Apple* por diversas razões, e, obviamente, o mesmo se aplica ao *S-Patch*. Concluindo, a ideia é desenvolver um sistema que capta informação de diferentes fornecedores de *wearables*, de modo a permitir a dar aos agentes de saúde outras perspetivas para ajudar os pacientes.

³ Abstração de um conjunto de serviços de vários projetos com o objetivo de fornecer uma funcionalidade genérica.

⁴ *Electronic Health Record*, em português Registo Eletrónico de Saúde

⁵ Empresa sediada na Coreia do Sul, que fornece serviços e produtos tecnológicos, disponível em <http://www.samsung.com/pt/>.

⁶ Conferência que permite a apresentação de produtos inovadores, incluindo *wearables*, <https://www.ces.tech/>.

4.5 Sumário

Este capítulo teve como objetivo principal apresentar o problema que esta dissertação se propôs a resolver. Para isso, foi definido o tema principal deste projeto (*Patient Generated Health Data*) e, também, foi enquadrado com outro conceito importante: *wearables* (secção 4.1). Foram abordados os vários desafios que resultam da junção destes dois temas (secção 4.2), e apresentadas as soluções que já existem para atacar o problema (4.3) juntamente com o que a solução pretendida acrescenta de novo (secção 4.4).

Nos próximos capítulos, serão abordados vários aspetos relacionados com a solução implementada para resolver o problema. Sendo que o capítulo 5 apresentará as várias decisões de alto nível tomadas e o capítulo 6 abordará aspetos de mais baixo nível da solução, para começar a dar entender como a solução foi desenhada. O capítulo 8 apresentará os vários aspetos de implementação da solução e o capítulo 7 explicará a metodologia usada no desenvolvimento de software, bem como as operações de suporte (planeamento e reuniões).

Capítulo 5

Âmbito da solução

Neste capítulo são abordadas as várias decisões de alto nível tomadas, e que servem de base para a solução, consequentemente o capítulo está dividido nas seguintes secções:

- 5.1 - Justifica o contexto escolhido.
- 5.2 - Apresenta e justifica a escolha das várias plataformas de fontes de dados.
- 5.3 - Justifica a seleção do *wearable*.
- 5.4 - Explica como pode ser implementado o RGPD (já abordado na secção 3.3) na solução.
- 5.5 - Explica como as *standards* de segurança (já abordados na secção 3.2) podem ser considerados para a solução.

5.1 Escolha do(s) contexto(s) de aplicação

Para avaliar cada contexto foi levantado um conjunto de *wearables* para cada um. Começando pelo contexto de "Risco de queda", chegou-se à conclusão de que grande parte destes *wearables* funcionam como sistemas de socorro, ou seja, não conseguem prever quando o utilizador poderá cair. Existindo uma exceção à regra, o *wearable Kinesis QTUG* [19], este *wearable* consegue oferecer um conjunto de métricas que prevê quando o utilizador poderá, eventualmente, cair. O grande entrave deste aparelho é o facto de ser um sistema fechado, que não fornece nenhuma forma de usar as suas funcionalidades por terceiros.

No contexto da "Postura", existe uma maior oferta, em que é possível identificar dois tipos de *wearables*:

- uns conseguem avisar o utilizador através de pequenas vibrações ou avisos para o telemóvel quando este está com uma postura incorreta, como por exemplo *UpRight Go* [53], *Alex Posture* [54], *Zitko Walk* [55] e *Lumo Lift* [56];
- outros funcionam como treinadores, ou seja, através de exercícios treinam a postura do utilizador, como por exemplo *Valedo* [22].

Nesta área existe apenas um *wearable* que fornece serviços para utilizar as suas funcionalidades: o *Lumo Lift* [57].

Relativamente ao contexto de "Diabetes", foi possível identificar três *wearables*: *Dexcom G5* [8], *K'Watch Glucose* [58] e *The Freestyle Libre Sensor* [59]. Estes aparelhos conseguem medir o nível de glucose do utilizador através de um simples contacto com a pele. Destes aparelhos, apenas um fornece uma forma de usar as suas funcionalidades: o *Dexcom G5* [60], sendo que só oferecem a possibilidade de aceder a dados de utilizadores que estejam nos EUA.

Sobrando dois contextos ("Caminhada" e "Sono"), antes de listar os *wearables* nestas áreas, é importante perceber que todos os *wearables* levantados que oferecem métricas sobre caminhada, também oferecem métricas relativas à qualidade de sono dos utilizadores. Nestes contextos, grande parte dos fornecedores oferecem APIs para usar os dados recolhidos pelos aparelhos. Devido à grande oferta, em vez de listar *wearables* (como foi feito para os outros contextos), vai-se listar alguns dos fabricantes mais populares (como a figura 5.1 mostra [61]):

- *Fitbit* [14];
- *Garmin* [15];
- *Google Fit* [62] (não fabrica *wearables*, mas tem um conjunto de *wearables* que se conectam com aplicação diretamente, explicado em mais detalhe na secção 5.2);
- *Samsung* [18];
- *Apple* [48].

É importante referir que a *Google Fit* não fabrica *wearables*, mas contém um grande número de utilizadores ativos, estando na posição número 4 com 2.1 milhões de utilizadores por mês desde de Julho de 2017 nos EUA, como a figura 5.2 demonstra [63].

De todas as áreas estudadas, é possível entender que a área "Caminhada/Sono" é a mais aberta ao desenvolvimento e com uma maior oferta de *wearables*, sendo assim foram estes os contextos escolhidos.

De seguida, ver-se-á estes contextos em mais detalhe, para se perceber o tipo de oferta que existe no mercado vão ser estudadas as funcionalidades de três aplicações: *Fitbit* [64], *Google Fit* [65] e *Samsung Health* [66].

A aplicação da *Fitbit* [64] oferece cinco módulos principais:

- Painel principal - contém as funcionalidades base.
- Desafios - desafia o utilizador a atingir certas metas (distância, calorias, entre outros).
- Orientação - oferece orientação durante os exercícios.
- Amigos - cria competição entre o utilizador e amigos.
- Notificações - notifica/altera o utilizador sobre certos estados (distâncias percorridas, etc).

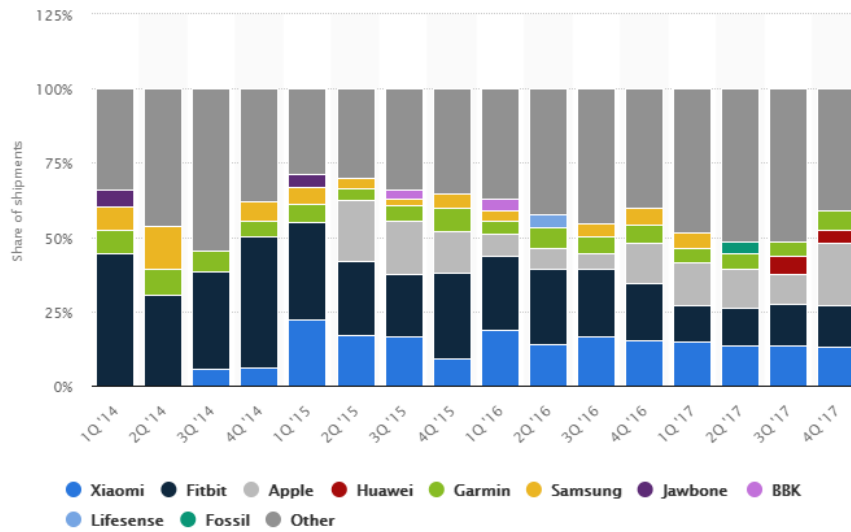


Figura 5.1: Quota de mercado de fornecedores de *wearables* entre 2014 e 2017

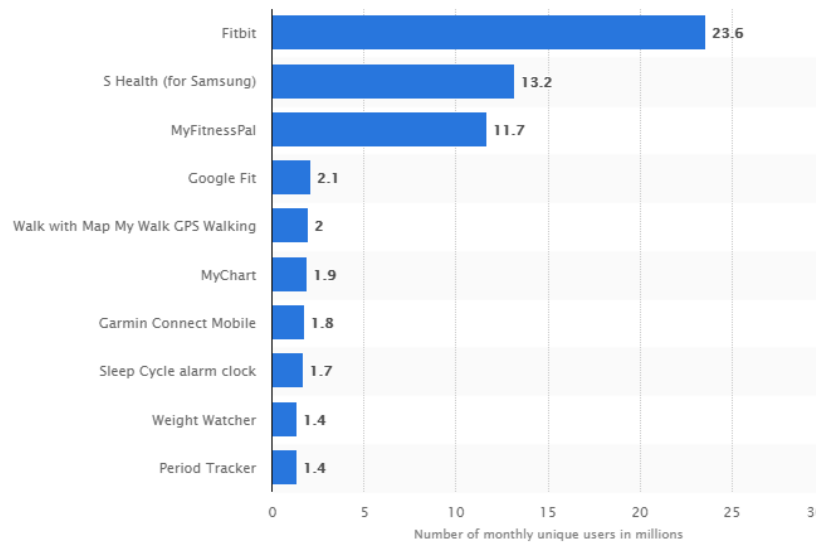


Figura 5.2: Aplicações de saúde/*fitness* mais populares nos EUA (com número de utilizadores ativos por mês) desde Julho de 2017

No painel principal existem as seguintes métricas:

- número de passos;
- número de degraus subidos;
- distância percorrida;

- calorias gastas;
- número de minutos ativos;
- "sono";
- peso;
- altura;
- batimento cardíaco;
- exercícios;
- comida/alimentos ingerida/os;
- litros de água bebidos.

É possível, também, ver certas métricas em detalhe, como por exemplo ao selecionar um exercício são mostradas as calorias queimadas, distância, batimento cardíaco médio, entre outros.

De todos estes conceitos existe um que é tratado de forma diferente: "sono". Através de uma combinação do movimento e da variação do batimento cardíaco do utilizador é possível verificar se o utilizador está a dormir. Durante o sono é possível identificar três tipos de estágios:

- Leve - acontece nas primeiras horas de sono, promove recuperação física e mental. O utilizador é acordado facilmente deste estágio de sono e o batimento cardíaco começa a diminuir.
- Profundo - permite recuperar melhor, ocorre normalmente depois do sono leve. É mais difícil de acordar deste estágio e o batimento cardíaco é menor nesta fase;
- REM - acontece depois do primeiro sono profundo. Neste estágio é onde os sonhos costumam ocorrer, sendo que o batimento cardíaco começa a aumentar e a ficar mais irregular.

A aplicação da *Fitbit* mede cada estágio de sono e calcula a percentagem de cada um dos três estágios, bem como a percentagem de tempo em que o utilizador teve acordado (figura 5.3). [67]

Relativamente à *Google Fit*, também é possível ter acesso a métricas como número de passos, distância percorrida, batimento cardíaco, exercícios, definir objetivos, etc. [65]

Por fim, *Samsung Health* permite reconhecer automaticamente exercícios, definir objetivos, ser desafiado por amigos, também oferece métricas como as da *Fitbit* (distância, batimento cardíaco, calorias, etc.). [66]

5.2 Plataformas de fontes de dados

Para além dos *wearables* como fontes de recolha de dados, também existe outra tipo de fonte de dados: dados introduzidos diretamente por utilizadores, como por exemplo peso, altura, nível de atividade física, entre outros. Este tipo de dados pode ser introduzido em aplicações, como por exemplo *Google Fit* [62].

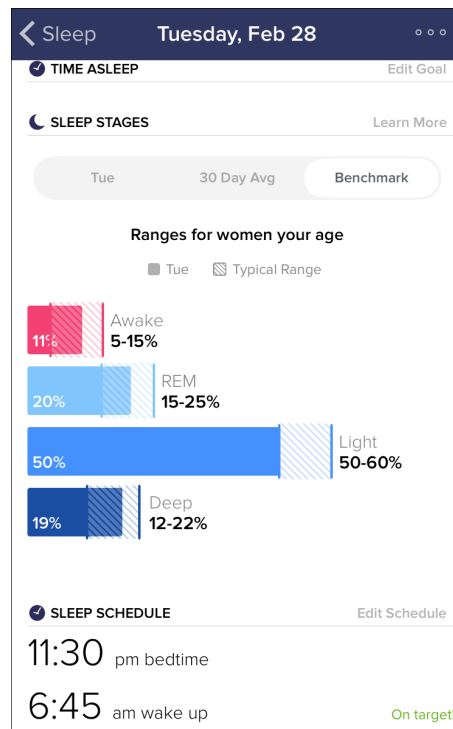


Figura 5.3: Estágios de sono na aplicação da *Fitbit*

Sendo assim, é necessário perceber que nem todos os dados são obtidos através dos *wearables*, ou seja, nasce uma oportunidade, para além de não consumir apenas dados recolhidos pelos *wearables*, também consomem dados introduzidos pelos utilizadores. E como os dados são recolhidos e armazenados numa plataforma, também abre portas a outra oportunidade, que é o facto de várias plataformas permitirem a incorporação de vários aparelhos *wearable*. Isto é, consumindo dados disponíveis numa plataforma, estão a usar-se dados de diferentes *wearables* que recolhem tipos de informação distinta.

Com esse objetivo em mente, foram estudadas quatro plataformas (dos fornecedores apresentados na secção 5.1, faltando apenas a *Garmin*, a qual foi excluída porque a sua API tem um custo associado [68]):

- *Google Fit* [62]
- *Fitbit* [14]
- *Apple Healthkit* [49]
- *Tizen (Samsung)* [69]

5.2.1 *Google Fit*

Google Fit [62] é uma aplicação desenvolvida para *Android*, mas também suporta uma versão Web. De um modo geral, a *Google Fit* é uma aplicação desenvolvida pela *Google* que permite

registar atividades de exercício físico e histórico de peso, também consegue contar número de passos, calorias queimadas, distância percorrida, entre outros. Para além de fornecer estas funcionalidades base também possui um grande conjunto de parceiros, permitindo a incorporação dos seus produtos com a *Google Fit* através da partilha dos dados. De seguida, são apresentados alguns exemplos de aplicações que funcionam com a *Google Fit*:

- *Lose It!* [70];
- *Under Armour Record* [71];
- *RunKeeper* [72];
- *Nike Run Club* [73];
- *MyFitnessPal* [74];
- *Instant Heart Rate by Azumio* [75];
- *Nokia Health Mate* [76];
- *SmartBand 2 SWR12* [77].

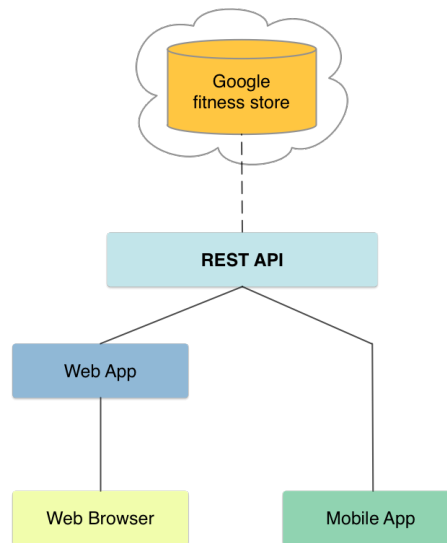
Por outro lado, é possível conectar a aplicação diretamente com um conjunto de *wearables*, que possuem como base o sistema operativo *Google Wear* [78]:

- *Fossil Q Control* [79];
- *Huawei Watch 2* [80];
- *New Balance RunIQ* [81];
- *LG Watch Sport* [82];
- *Polar M600* [83];
- *Moto 360 Sport* [84].

Para utilizar os dados capturados pela *Google Fit*, a *Google* fornece duas alternativas:

- *Fit API para Android* – APIs que permitem utilizar um conjunto de serviços para desenvolver aplicações *Android*. [85]
- *Fit REST API* – permite guardar e aceder informação guardada na base de dados da *Google Fit* através de uma WEB API. [86]

Ambas as soluções ligam-se á base de dados da *Google*, que neste caso se chama *fitness store*, como a figura 5.4 demonstra.

Figura 5.4: *Google Fit* arquitetura de alto nível

5.2.2 *Fitbit*

Fitbit [14] é diferente da *Google Fit*, de um modo geral a *Fitbit* é uma empresa que desenvolve produtos *fitness*, ou seja, *wearables* e acessórios focados a oferecer métricas aos utilizadores de modo a tentar melhorar o seu desempenho físico, bem como o seu estilo de vida. Para além dos aparelhos desenvolvidos, também oferece uma aplicação para *Android*, *iOs* e *WEB*.

A oferta da *Fitbit* para utilizar os dados armazenados do seu lado é idêntica à oferta da *Google* (secção 5.2.1), oferecendo também duas alternativas:

- *Fitbit SDK* [87] - permite desenvolver aplicações e ecrãs para os relógios através do *Fitbit Studio*. [88]
- *Fitbit Web API* - permite aceder aos dados armazenados na base de dados da *Fitbit* através de uma *WEB API*. [89]

5.2.3 *Apple Healthkit*

Como já foi referido no capítulo 4.3, *Apple Healthkit* é uma *framework* que permite que aplicações de saúde/*fitness* comuniquem entre si com o objetivo de partilhar informação.

A oferta da *Apple* é mais fechada à própria tecnologia *Apple*, enquanto que a *Google* e a *Fitbit* oferecem uma *Web API* que pode ser usada em qualquer tecnologia, a *Apple* apenas fornece o *Healthkit* para ser utilizado em aplicações *iOs* e *watchOS*. [90]

5.2.4 Tizen/Samsung

Tizen [69] é diferente de todas as soluções anteriores (5.2.1, 5.2.2 e 5.2.3), devido ao facto de ser um sistema operativo de código aberto¹, foi desenvolvido pela *Linux Foundation* [91] e contém vários perfis²:

- *Tizen IVI* - veículos.
- *Tizen Mobile* - telemóveis;
- *Tizen TV* - televisões;
- *Tizen Wearable* - wearables.

É importante referir que a *Tizen* é usado como sistema operativo dos wearables da *Samsung* [18]. *Tizen* oferece duas alternativas para desenvolver/contribuir para plataforma:

- Permite desenvolver aplicações (equivalente ao *Android* e *iOs*). [92]
- Permite contribuir para plataforma, desenvolvendo funcionalidades para o sistema operativo *Tizen*. [93]

Como as APIs da *Tizen* são fechadas ao sistema operativo, foi encontrado uma alternativa fornecida pela *Samsung* que permite aceder as dados captados por wearables via WEB API [94]. Todos os dados captados pelos aparelhos estão disponíveis na aplicação *Samsung Health* [95].

5.2.5 Escolha das plataformas

Pela leitura da secção 5.2, consegue-se chegar à primeira exclusão: *Apple Healthkit*. Como é uma solução que permite ser só usada em aplicações *iOs*, faz com que o desenvolvimento fique limitado a essa plataforma. Sobrando três fontes de dados: *Google Fit*, *Fitbit* e *Samsung*. Destas três possíveis plataformas, todas permitem desenvolver sem que o sistema fique ligado apenas a uma plataforma, ao contrário da solução oferecida pela *Apple*. A escolha podia cair nestas três fontes de dados, o problema é que a solução oferecida pela *Samsung* necessita de uma permissão antes que a sua solução possa ser utilizada. Depois da aplicação [96] ser submetida à *Samsung*, a resposta não foi obtida no tempo de vida útil do projeto, sendo assim a opção da *Samsung* foi excluída. Chegou-se à conclusão que as plataformas selecionadas são: *Google Fit* e *Fitbit*.

5.3 Escolha do wearable

Com o contexto escolhido só falta escolher o *wearable* que vai ser usado para teste, das duas fontes de dados só foi escolhido um *wearable* de uma fonte de dados, essa fonte de dados foi a

¹Modelo de desenvolvimento em que o código é público, permitindo o seu desenvolvimento por vários colaboradores.

²Perfil, neste caso, refere-se a uma especialização/tipo do sistema operativo.

Fitbit. *Fitbit* oferece uma gama grande de wearables, os principais wearables são: *Ionic*, *Blaze* e *Charge 2* [97]. Antes de anunciar a escolha é necessário perceber as principais diferenças entre os aparelhos.

Em termos de design os aparelhos são diferentes, se bem que o *Ionic* e o *Blaze* são sensivelmente parecidos. *Charge 2* é uma pulseira inteligente com o design mais discreto (como a figura 5.7 mostra), os restantes são mais parecidos com um relógio (figura 5.5 e 5.6).

Figura 5.5: *Blaze Fitbit*Figura 5.6: *Ionic Fitbit*Figura 5.7: *Charge 2 Fitbit*

De um ponto de vista de funcionalidades básicas, todos contam passos, calorias, distância, escadas, entre outros [98]. Primeiro, vai ser feita uma comparação entre a *Blaze* e a *Charge 2*. Entre estes dois aparelhos existem duas diferenças, o *Blaze* oferece exercícios no ecrã e permite controlar música diretamente sem que o utilizador necessite de ir ao telemóvel. Destes dois wearables o mais completo é o *Blaze*, assim o *Blaze* vai servir de base para comparar com o *Ionic*. As funcionalidades que o *Ionic* oferece em comparação ao *Blaze* são as seguintes:

- pode ser usado na piscina (é resistente a 50 metros abaixo da água);
- tem GPS (não é preciso o uso de telemóvel);
- oferece aplicações populares para serem usadas através do ecrã do aparelho;
- guarda e reproduz música;
- tem NFC, que permite guardar e usar cartões de crédito e débito para fazer pagamentos.

Destes três wearables é fácil perceber que o mais completo é o *Ionic*, seguindo-se o *Blaze* e por fim o *Charge 2*. Mas é importante referir que existe outro critério de igual importância que é o preço, sendo o preço proporcional às funcionalidades oferecidas. O preço numa escala decrescente é o seguinte:

1. *Ionic* (349,95 euros);
2. *Blaze* (199,95 euros);
3. *Charge 2* (159,95 euros).

Pesados todos os fatores a escolha recai no *Charge 2*, devido ao facto de ter todas as funcionalidades necessárias para esta prova de conceito e, também, é o aparelho com um preço menor.

Inicialmente o protótipo deveria ser testado com dados reais de um *wearable*, mas devido a problemas alheios ao contexto do projeto, o aparelho não chegou a tempo para ser usado para teste. Sendo assim, foram criados dados de teste para cada uma das fontes de dados.

5.4 RGPD

Como é possível perceber na secção 3.3, o RGPD é um processo bastante complexo, mas devido ao facto de ser aplicado neste ano corrente e, também, devido á natureza dos dados usados ser sensível (dados de saúde), fica a ser obrigatório abordar este tema. Para implementar devidamente o RGPD em qualquer projeto é necessário uma revisão total do projeto, resultando num novo conjunto de requisitos, o que faz com que o impacto de tais alterações fique dependente do tamanho/complexidade do sistema em questão. Num sistema novo (como é o caso deste projeto) esta implementação pode se tornar mais acessível, mas, mesmo assim, depois de o produto ser implementado e lançado, o *feedback* dos utilizadores é um aspeto importante para se perceber se eles sentem se a sua informação estar a ser tratada de forma indevida. Ou seja, para ser possível ter o RGPD devidamente implementado faz com que leve a um processo difícil e moroso.

Através da leitura da secção 3.3 é possível extrair algumas *guidelines* importantes:

- A partir do momento que a informação chega a aplicação, quem a desenvolve é totalmente responsável pela sua segurança.
- É necessário permitir ao utilizador exportar os dados para um formato digital, e se possível diretamente para outros sistemas.
- O utilizador tem o direito de apagar a sua informação a qualquer momento.
- É importante que o utilizador autorize todas as transações de dados entre os vários sistemas.

Com estas *guidelines* é possível chegar à definição de dois requisitos (presentes na secção 8.2.1): *REQ-FUNC6* e *REQ-FUNC7*. Foram identificados estes requisitos, mas podiam resultar em muitos mais, o objetivo é apenas mostrar que o RGPD afeta e vai continuar afetar muitas soluções que já estão no mercado, bem como as que estão para vir.

Para terminar, é importante abordar o uso de telas de consentimento, as telas de consentimento permitem dar a entender ao utilizador, na perfeição, o que se está a fazer com a informação dele, servindo para pedir permissão antes de cada ação principal relacionada com a manipulação de dados. Por exemplo, na primeira vez que se pede para importar os dados do utilizador de uma das fontes de dados, uma tela deve sugerir a pedir permissão para efetuar essa operação.

5.5 Standards de segurança

Como se trata de uma aplicação que lida com dados considerados sensíveis, é importante ter a noção de que esses dados tem que estar seguros não só para estarem de acordo com a lei (RGPD 3.3), mas também para oferecer segurança e confiança aos utilizadores. Por outro lado, é importante entender que como se trata de uma prova de conceito não é considerado um objetivo para ser implementado de imediato, embora é um aspeto relevante que necessita de ser abordado (secção 3.2). Ou seja, é importante levantar o estado de arte na área da segurança da informação, mas não foi considerado um requisito a ser implementado.

Capítulo 6

Desenho da solução

Este capítulo dá seguimento ao capítulo 5, em que é:

- Explicado, em mais detalhe, o processo de pensamento por detrás do estudo do modelo de dados de cada uma das fontes (secção 6.1);
- Apresentado o modelo de dados do projeto resultante da junção do modelo das fontes (secção 6.2).
- Estudado o impacto da adição de uma nova fonte de dados, funcionando como processo de validação da qualidade da solução (secção 6.3).
- Apresentado táticas para reduzir o impacto da alteração do modelo de dados implementado na base de dados (secção 6.4).
- Apresentada a solução arquitetural, de alto nível, do sistema implementado (secção 6.5).

6.1 Estudo ao modelo de dados das fontes escolhidas

Nesta secção é estudado o modelo de dados de cada uma das fontes escolhidas: *Google Fit* e *Fitbit*. Antes de entrar no estudo de cada modelo, é importante perceber conceitos transversais a cada uma das APIs.

Todas as APIs são REST e usam o protocolo HTTP. REST [99] é um estilo arquitetural que insere um conjunto de restrições nas APIs que usam o HTTP:

- Arquitetura cliente-servidor;
- Não guarda estado;
- Permite o uso de *cache*;
- Arquitetura em camada.

O protocolo HTTP [100] permite que sistemas comuniquem através da Internet, definindo um conjunto de métodos padrão, entre os quais se destacam:

- *GET* - lista valores;
- *DELETE* - apaga valores;
- *POST* - introduz novos valores;
- *PUT* - atualiza valores.

Para além de terem em comum o protocolo HTTP e o estilo arquitetural REST, também tem por base a mesma forma de autorização. Para ter acesso aos dados do utilizador de outra plataforma, uma das formas menos seguras seria o utilizador (das plataformas da *Google Fit* e *Fitbit*) introduzir os dados de acesso no lado da aplicação a ser desenvolvido neste projeto. Para colmatar essa fragilidade surgiu um protocolo de autorização denominado por O-Auth. O-Auth já possui duas versões (1.0 e 2.0), a que vai ser utilizada neste projeto é a versão 2.0. [101] O-Auth 2.0 tem vários perfis/tipos:

- *Authorization Code*;
- *Implicit*;
- *Password*;
- *Client Credentials*;
- *Device Code*;
- *Refresh Token*.

O tipo usado nesta prova de conceito foi o *Authorization Code*. Esta especificação do protocolo consiste no seguinte: o cliente (sistema desenvolvido no projeto) tem um identificador e um segredo, juntamente com o um *url* (que vai servir para a fonte de dados redirecionar para o cliente) e o *scope* (o que se pretende consumir do outro lado, por exemplo exercícios físicos, batimento cardíaco, entre outros).

É feito um pedido com estes dados, de seguida o sistema redireciona o utilizador para um a janela de autenticação fora do controle do sistema desenvolvido neste projeto. Esta janela é totalmente controlada pela a fonte de dados a qual se está a pedir a permissão para usar os dados. Com a autorização dada pelo utilizador, a fonte de dados retorna um *token* de acesso (conjunto de caracteres), permitindo que o sistema tenha acesso à informação autorizada pelo utilizador. [101]

6.1.1 *Google Fit*

Google Fit tem uma abordagem considerada muito desacoplada e modular, ou seja, para obter o valor é preciso o tipo de dados ¹ (peso, altura, etc) e o intervalo de tempo (data de inicio e fim). Como por exemplo, se o objetivo for obter o valor do batimento cardíaco no dia 03 de Fevereiro

¹O termo *tipo de dados* refere-se ao conceito *data source* usado na documento oficial da Google [86] (para facilitar a leitura).

de 2017 às 17:33h até ao dia 04 de Fevereiro de 2017 às 16:03h, basta fazer um pedido com o tipo de dados do batimento cardíaco nesse intervalo de tempo que o valores serão retornados [102]. A lista dos tipos de dados mais importantes que existem é a seguinte:

- exercícios;
- calorias;
- distância;
- batimento cardíaco;
- peso;
- altura;
- localização;
- velocidade;
- número de passos.

Para obter o peso do utilizador é feito um pedido com o intervalo de tempo que se pretende, associado com o tipo de dados de "peso". Quando o conceito é simples como o peso, altura, batimento cardíaco a leitura é acessível. Por outro lado, quando o resultado pretendido é o output de um exercício físico, a abordagem não é tão linear.

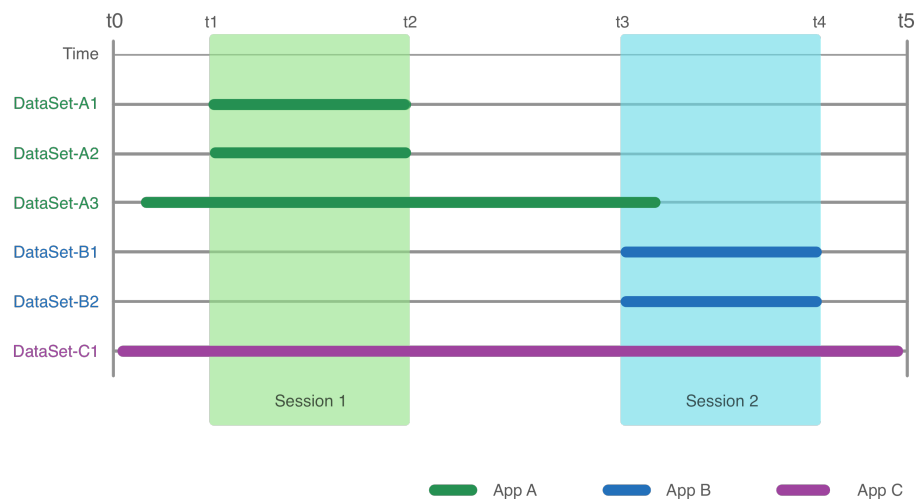


Figura 6.1: Conceito de sessão na *Google Fit*

Através da leitura da imagem 6.1 [103] é possível perceber que existem dois exercícios físicos: "Session 1" e "Session 2". Ou seja, para obter a distância (ou outro output da sessão, como por exemplo calorias, passos, entre outros) de um exercício físico é preciso, numa primeira fase, fazer um pedido com o tipo de dados de "sessão" para listar todas as sessões existentes [102].

Numa segunda fase, com os valores de início e fim de cada sessão, faz-se um pedido usando o tipo de dados "distância" com o tempo de início e fim da sessão, para obter a distância nesse exercício. É importante ainda referir que, se um conjunto de dados ("dataset") for introduzido no intervalo de tempo que engloba o intervalo de tempo da sessão, esse conjunto de dados faz parte da sessão no intervalo de tempo da mesma. Ou seja, "DataSet-C1" tem um tempo de vida de "t0" a "t5", mas de "t1" a "t2" este conjunto de dados está associado à sessão "Session 1" e de "t3" a "t4" à sessão "Session 2". Nos outros intervalos de tempo ("t0" a "t1", "t2" a "t3" e "t4" a "t5") são valores normais associados a medidas normais, sem estarem ligados a nenhum exercício físico.

Para finalizar, é importante referir que para além de listar os valores também é possível introduzir valores em tipos de dados que foram criados pela mesma fonte. Ou seja, se um tipo de dados foi criado pela aplicação "X", só essa pode introduzir valores nesse tipo de dados. Com isto é possível deduzir que se a aplicação "X" tentar introduzir valores no tipo de dados "peso" criado pela aplicação da *Google Fit*, tal não vai resultar, devido ao facto de o tipo de dados em questão ser um valor padrão criado pela aplicação *Google Fit*. Concluindo, não é possível introduzir dados que sejam visíveis na aplicação e na versão web da *Google Fit*, através da REST API.

6.1.2 *Fitbit*

A abordagem da *Fitbit* é diferente da *Google Fit*, a *Fitbit* apresenta um modelo mais estruturado, dividindo o modelo em sete categorias principais [89]:

- Exercícios - exercício físico e os resultados associados.
- Corpo e Peso - valores corporais (massa gorda, peso e índice de massa corporal).
- Aparelhos - aparelhos *wearables* que o utilizador possui.
- Nutrição - alimentos ingeridos, quantidades, entre outros.
- Amigos - listagem dos amigos que o utilizador possuiu na *Fitbit*.
- Batimento cardíaco - varias abordagens para obter e trabalhar com o dados de batimento cardíaco.
- Sono - dados relacionados com a qualidade de sono (explicado na secção 5.1).

Dentro de cada categoria existe uma listagem de métodos HTTP que permitem manipular a informação desse contexto. Por exemplo, na categoria dos "exercícios" [104] existem as seguintes formas de aceder e trabalhar a informação:

- Obter o resumo diário do exercício físico - resumo e listagem dos exercícios físicos do utilizador num dado dia.
- Valores temporais - dados temporais num intervalo de tempo referentes a um determinado recurso do exercício (passos, calorias, distância).

- Introduzir exercícios - permite introduzir novos exercícios.
- Tipos de atividade - permite listar os vários tipos de atividade (corrida, caminhada, entre outros).
- Objetivos - permite listar os objetivos definidos (número de passos, distância, etc) pelo utilizador.
- Estatísticas - permite obter estatísticas gerais de todos os exercícios num dado intervalo de tempo.

6.2 Modelo de dados

Para perceber melhor o resultado final do modelo de dados do sistema é importante perceber a linha de pensamento que levou a esse modelo. Pela leitura e análise da secção 6.1 é possível perceber que as duas fontes de dados fornecem os dados de forma distinta.

Através da documentação da API da *Fitbit* [89] é possível chegar a um modelo de dados (presente nos anexos, secção A.2). Com o modelo de dados da *Fitbit*, foi feita uma análise aos dados fornecidos e excluídos alguns dados considerados desnecessários. De seguida, foi analisado o modelo da *Google Fit*, ao qual levou a uma abordagem diferente.

Enquanto que na *Fitbit* através da análise da documentação era possível chegar a um modelo estruturado, na *Google Fit* não foi possível, a abordagem passou por perceber que tipo de dados existem e perceber que valores se podiam guardar (também presente nos anexos secção A.1).

É importante perceber que as duas plataformas possuem tipos de dados diferentes e mesmo assim é necessário chegar a um modelo final. Por exemplo, a *Fitbit* pode ter o dado X e Y, mas a *Google Fit* só ter o dado Y, o que acontece é que não se vai desperdiçar o dado X só porque a *Google Fit* não o tem. Como solução a este problema consideram-se ambos os dados (sempre que fizer sentido para o contexto do projeto), sendo que se a leitura é feita à API da *Google* o resultado final será: dado Y fica com um valor e o dado X fica vazio. O que pode levar a que utilizadores com fontes de dados diferentes tenham uma visualização diferente dos dados.

Relativamente aos dados da qualidade de sono, apenas utilizadores da *Fitbit* tem acesso, enquanto que os da *Google* não tem acesso, devido ao facto de a *Google* não fornecer este tipo de dados através da sua API. O exemplo anterior é referente a uma categoria principal de dados (*Sono*), também existe outro exemplo que se relaciona com outra categoria: *Valores corporais*. A *Fitbit* fornece quatro valores (índice de massa corporal, peso, altura e massa gorda), enquanto que a *Google Fit* fornece apenas dois (altura e peso). Neste caso, quando o utilizador for visualizar os dados na aplicação móvel, se possuir um *wearable* da *Fitbit* tem acesso às quatro métricas, por outro lado se tem um aparelho da *Google Fit* visualiza só os outros dois dados. De um ponto de vista da base de dados não muda nada, o que muda é a forma como a aplicação móvel mostra os dados, sempre que verifica um dado com valor zero ou vazio, simplesmente omite esse dado. Assim, o sistema fica independente do tipo de fonte de dados do utilizador.

Para melhor perceção do modelo dados, este foi dividido em cinco categorias:

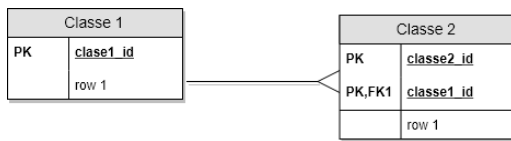


Figura 6.2: 1 para muitos



Figura 6.3: 1 para 1

- *Utilizador*: contém os dados do utilizador, aparelhos que possui e configurações das plataformas.
- *Valores corporais*: contém métricas relativas a valores corporais do utilizador, como peso, índice de massa corporal, massa gorda e altura.
- *Batimento cardíaco*: dados relativos ao batimento cardíaco, pode ser de dois tipos: leitura de um valor (valor com a data em que foi lido, apenas) e durante um exercício físico (com zonas de batimento cardíaco).
- *Exercício físico*: output do exercício físico (que pode ser caminhada ou corrida e tem métricas como número de passos, distância, etc).
- *Sono*: informação relacionada com a qualidade de sono.

Antes de entrar na análise do modelo de dados é importante perceber que foi usado um diagrama de base de dados ², em que existem dois tipos de ligações:

- 1 para muitos - significa que uma entrada na tabela "classe 1" corresponde a uma e só uma entrada na tabela "classe 2", mas uma entrada na tabela "classe 2" pode ter uma ou várias entradas na tabela "classe 1" (figura 6.2).
- 1 para 1 - significa que uma entrada na tabela "classe 1" corresponde a uma e só uma entrada na tabela "classe 2" e vice versa (figura 6.2).

6.2.1 Utilizador

Através da análise da imagem 6.4, é possível identificar a tabela do utilizador (*User*) que possui dados relativos ao utilizador. Dependente dessa tabela estão as tabelas:

- *Aparelho (Device)* - informação relativa ao aparelho.
- *UserConfigurationFitbit* - configuração da API da *Fitbit*, inclui unidades de cada das grandezas usadas, bem como informação adicional que é útil.
- *UserConfigurationGoogleFit* - configuração equivalente à tabela da *Fitbit*, mas com configurações da *GoogleFit*.

²É importante perceber que cada linha da tabela é equivalente a um coluna na implementação da base de dados. Também, é preciso ter atenção que cada linha de uma tabela (coluna na implementação) tem o nome da linha com o respetivo tipo de variável. *Varchar* significa um conjunto de caracteres, *int* um número inteiro e *double* um número com casa decimais.

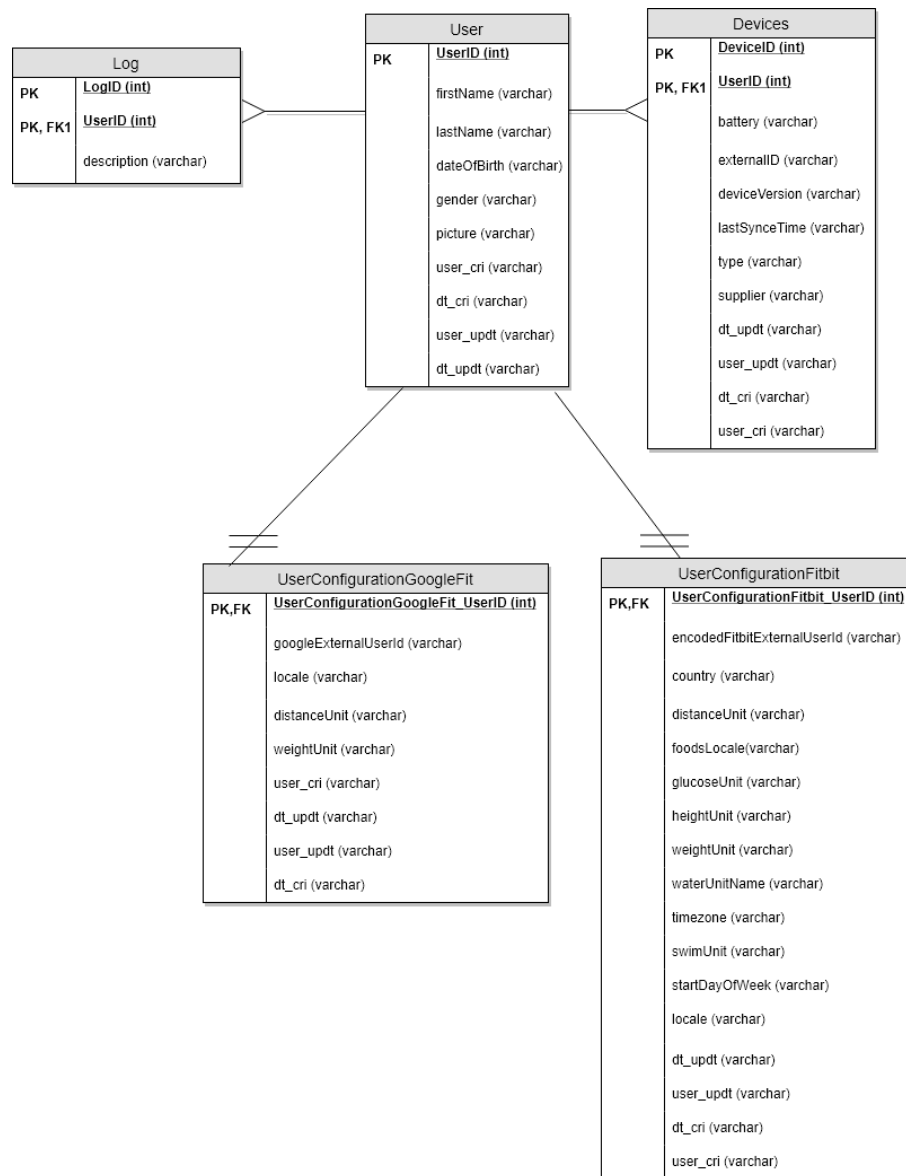


Figura 6.4: Modelo de dados - Utilizador

- *Log* - Tabela que permite guardar cada operação realizada pelo utilizador.

Ao analisar o diagrama (6.4) é possível perceber que cada utilizador pode ter 1 ou mais *devices* e o mesmo verifica para a tabela *log*. Em relação às configurações (*UserConfigurationFitbit* e *UserConfigurationGoogleFit*), o utilizador tem só uma configuração tanto para a *Fitbit* como para a *Google Fit*.

É importante referir que o sistema não foi implementado para ter um registo de utilizador, devido ao facto de ser um tópico em aberto que irá ser abordado numa fase posterior, fugindo ao âmbito desta dissertação.

6.2.2 Valores corporais

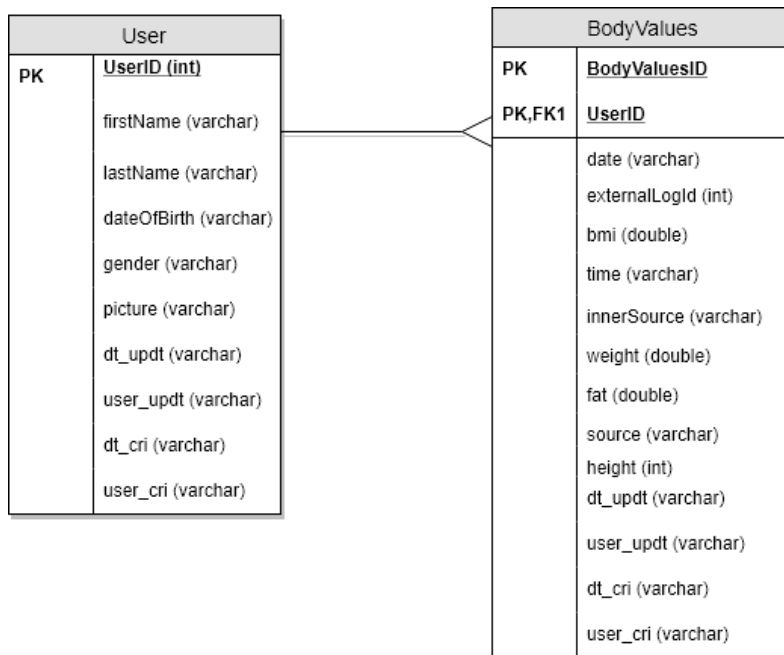


Figura 6.5: Modelo de dados - Valores corporais

Relativamente aos *Valores corporais*, existe uma tabela denominada por *BodyValues*, que contém métricas corporais, como o peso (*weight*), altura (*height*), índice de massa corporal (*bmi*) e massa gorda (*fat*).

6.2.3 Batimento cardíaco

No batimento cardíaco é necessário perceber que existem dois tipos:

- Leitura normal, corresponde à tabela *HeartRateValue*.
- Leitura durante um exercício físico, que contém zonas (tabela *HeartRateZone*).

Numa leitura normal tem a data (*dateTime*) em que foi lida com o valor da leitura (*heartRateValue*). Nas zonas tem o valor mínimo (*min*) e máximo (*max*), juntamente com os minutos (*minutes*) que essa zona teve. O atributo *restingHeartRate* é o batimento no qual o utilizador teve sem se mover durante um exercício físico.

6.2.4 Exercício físico

Relativamente aos exercícios físicos existem três outputs a destacar: distância percorrida (*distance*), calorias consumidas durante o exercício (*calories*), número de passos (*steps*) e a duração (*durationMilliseconds*).

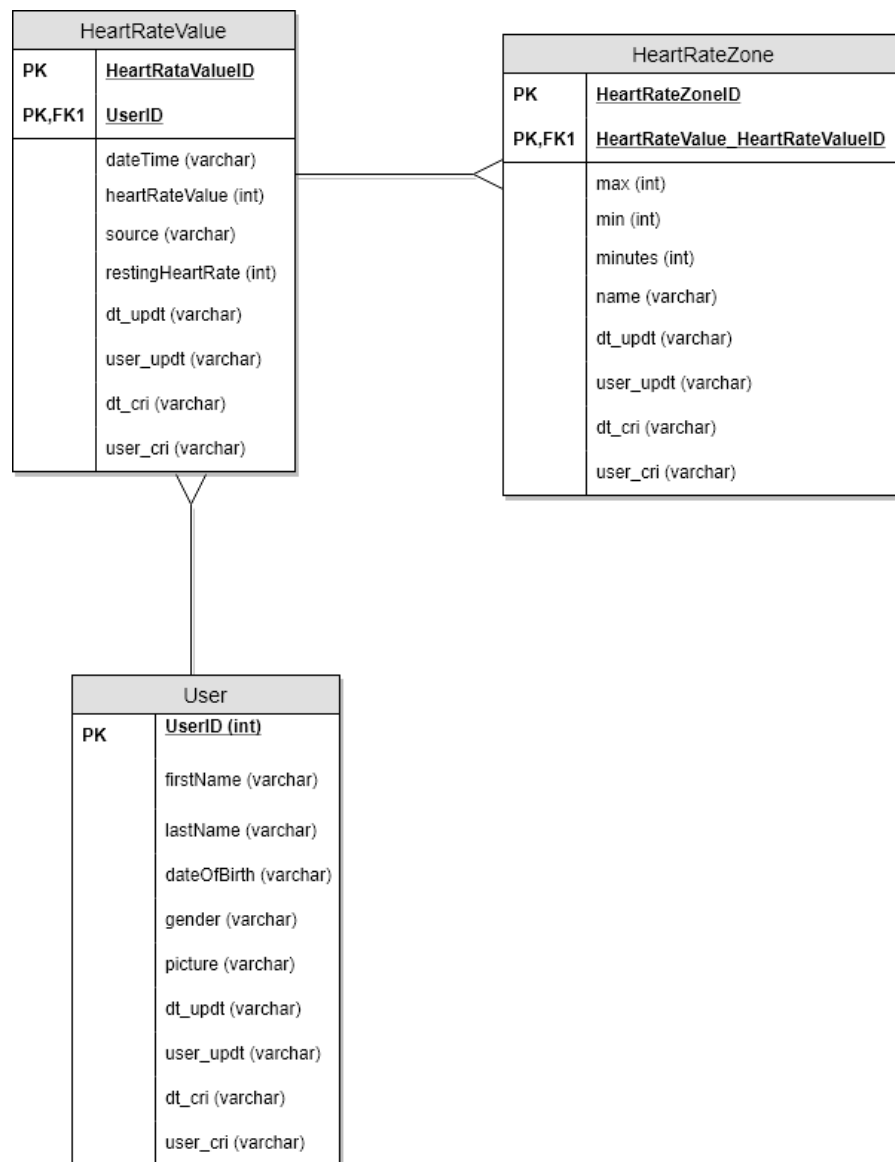


Figura 6.6: Modelo de dados - Batimento Cardíaco

6.2.5 Sono

O Sono é contexto com mais tabelas (figura 6.8):

- *Sleep*: tabela principal com métricas principais.
- *SleepData*: tabela com o sumário (estatísticas) de cada estágio de sono.
- *SleepTypeOfReading*: vários estágios de sono.

Para perceber bem o conceito de estágios de sono é importante ler a secção 5.1.

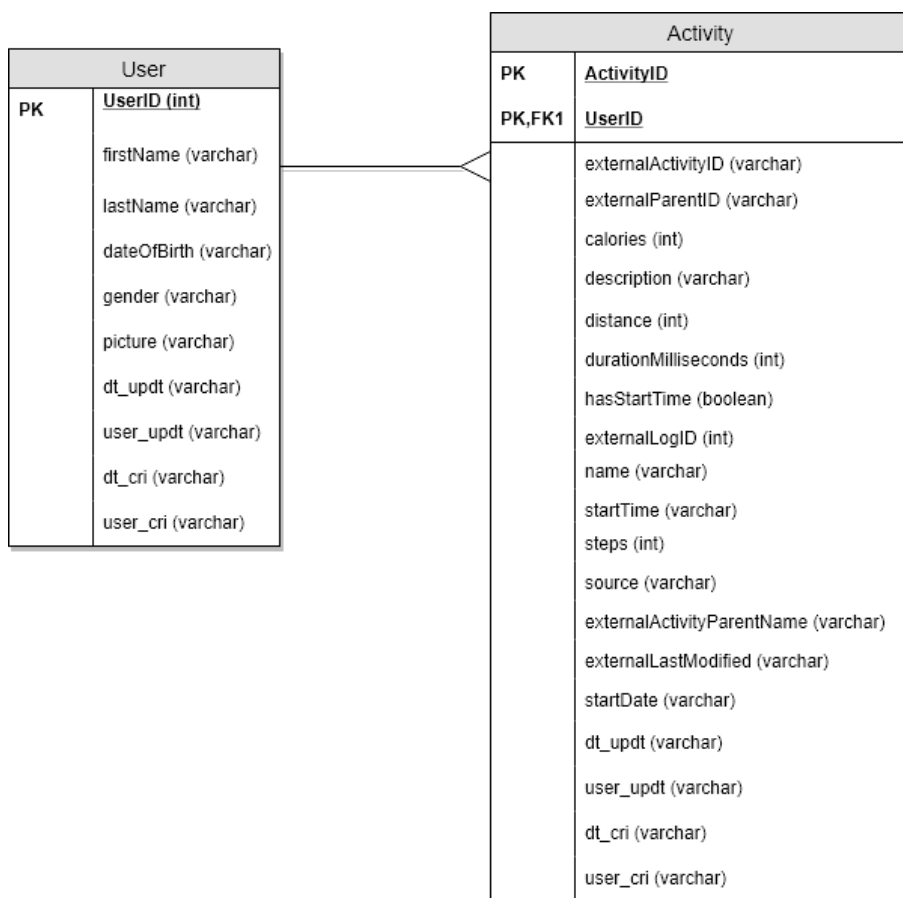


Figura 6.7: Modelo de dados - Exercício Físico

6.2.6 Notas finais

Nesta secção são apresentadas algumas notas finais para complementar as sub-secções: [6.2.1](#), [6.2.2](#), [6.2.3](#), [6.2.4](#) e [6.2.5](#). Na tabela [6.1](#) consegue-se perceber que tipo de informação é comum (ou não) entre as duas plataformas de fontes de dados.

Valor	<i>Fitbit</i>	<i>Google Fit</i>
<i>BMI</i>	X	
<i>Fat</i>	X	
<i>Weight</i>		X
<i>Height</i>	X	X
<i>HeartRateZone</i>	X	
<i>restingHeartRate</i>	X	
<i>heartRateValue</i>	X	X
<i>Activity</i>	X	X
<i>Sleep</i>	X	

Tabela 6.1: Comparação de informação do modelo de dados entre fontes

É importante ainda referir quatro aspetos:

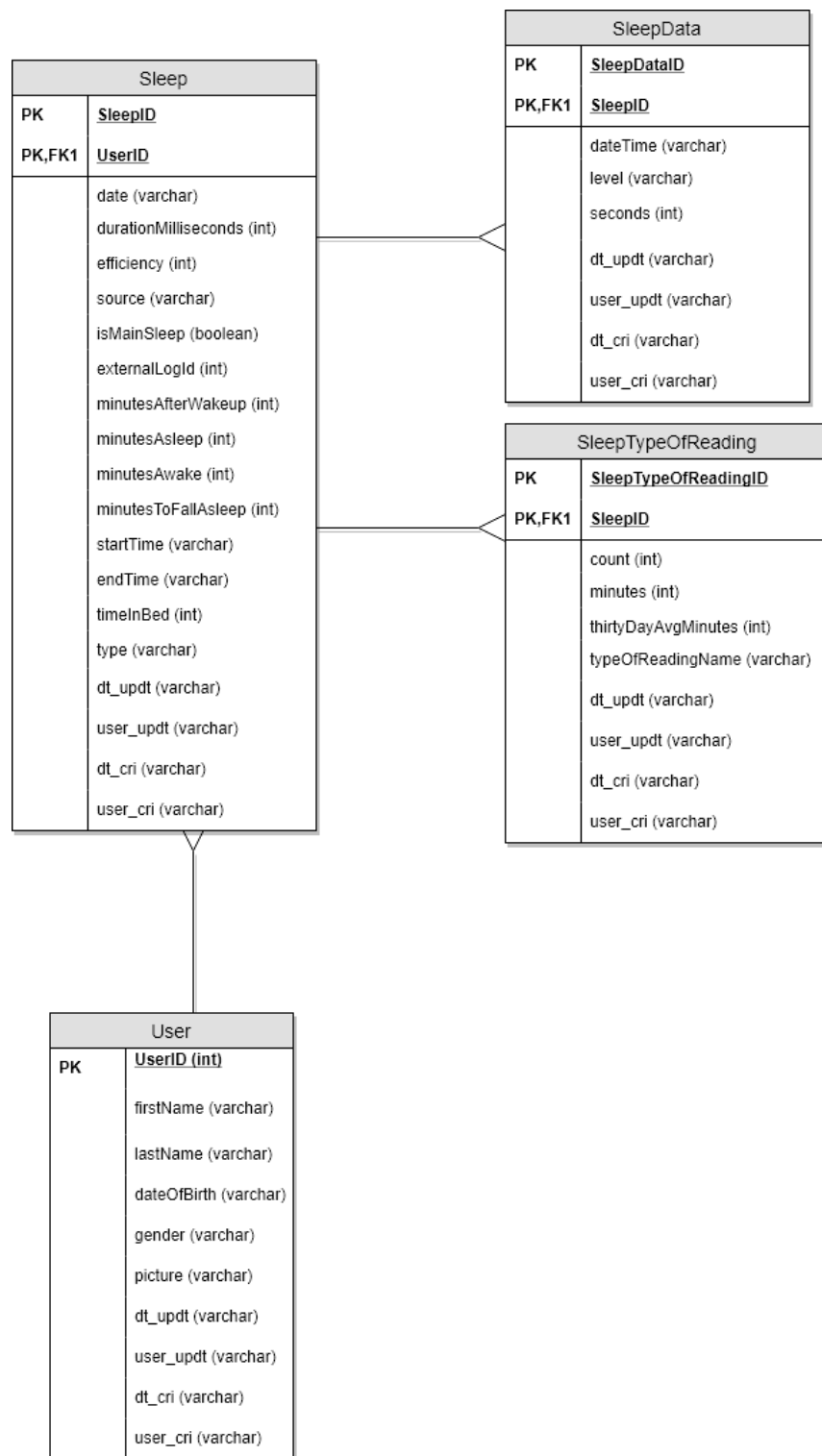


Figura 6.8: Modelo de dados - Sono

- Todos os atributos com *external* no nome são referentes a identificadores externos das fontes de dados.

- Todas as tabelas tem um atributo *date* ou *dateTime* que se refere a data (dia e hora) em que a leitura foi efetuada na respetiva fonte de dados.
- Cada tabela tem um atributo a identificar a fonte de dados (*source*).
- Todas as tabelas tem as linhas *dt_updt*, *user_updt*, *dt_cri* e *user_cri*, que são valores que complementam a tabela *Log* e permitem verificar quem criou e alterou a linha:
 - *user_cri* - Identificador de quem criou.
 - *dt_cri* - Data e hora de quando foi criado.
 - *user_updt* - Identificar de quem alterou.
 - *dt_updt* - Data e hora de quando foi alterado.

6.3 Caso de estudo - *Samsung Health Server SDK*

Esta secção tem como objetivo verificar o impacto que tem no sistema o acréscimo de uma nova fonte de dados, a fonte de dados em questão é a da *Samsung*, denominada por *Samsung Health Server SDK*. Analisando a documentação da *Samsung* [94], verifica-se que a abordagem da *Samsung* ao fornecer os dados é idêntica à da *Google Fit*, ou seja, também fornece os dados orientados a tipo de dados, que são os seguintes:

- temperatura ambiente;
- nível de glucose;
- pressão arterial;
- temperatura corporal;
- cafeína ingerida;
- informação relativa a um eletrocardiograma;
- exercício físico;
- escadas caminhadas;
- nutrição;
- hemoglobina (*hba1c*);
- batimento cardíaco;
- saturação do oxigénio no sangue;
- sono;
- estágio de sono;

- contagem de passos;
- água ingerida;
- peso;
- dados do utilizador;
- altura;
- aparelhos *wearable*.

Para introduzir esta nova fonte de dados é necessário duas etapas:

- Primeira etapa (secção 6.3.1) - fazer as alterações necessárias ao modelo de dados;
- Segunda etapa (secção 6.3.2) - desenvolver o novo controlador para tratar dos dados da *Samsung* (equivalente ao componente *FDictionary* e *GDictionary*, explicado em mais detalhe na secção 8.3).

6.3.1 Alterações à base de dados

Ao analisar o tipo de dados que a nova fonte de dados fornece, é preciso ter em atenção a dois aspetos:

- verificar se os tipos de dados podem-se encaixar no modelo de dados existente;
- verificar se existe novos dados que se podem incluir no modelo de dados.

Para verificar se os dados que estão representados no modelo de dados (secção 6.2) existem ou não na API da *Samsung*, vai ser estudada cada uma das categorias já apresentadas:

- *Utilizador*.
- *Valores corporais*.
- *Batimento cardíaco*.
- *Exercício físico*.
- *Sono*.

Relativamente aos dados do *Utilizador*, a *Samsung* fornece todos os dados necessários relativos à informação do utilizador [105], incluindo também uma lista dos aparelhos do utilizador [106]. Na categoria dos *Valores corporais*, contém os mesmo valores que a *Google Fit*: peso [107] e altura [105]. No *Batimento cardíaco*, fornecem as duas perspetivas: um único valor (atributo *heart_rate_count*) e zonas relativas ao exercício físico (*min*, *max* e *heart_rate_count*) [108]. Na perspetiva do *Exercício físico*, também disponibilizam os principais outputs: distância, passos

e calorias [109]. Por fim, na categoria do *Sono*, oferecem dados relativos ao sono [110] e aos estágios de sono [111].

Com esta análise é possível perceber que toda a informação presente no modelo existente é fornecida pela *Samsung*, à exceção dos valores do índice de massa corporal e massa gorda (à semelhança com a *Google*).

Na perspectiva de novos valores que possam ser adicionados ao modelo, podem existir três:

- temperatura corporal [112];
- saturação do oxigénio [113];
- eletrocardiograma [114].

Os dois primeiros valores (temperatura corporal e saturação do oxigénio) podem ser considerados novos valores corporais, que resulta em dois novos atributos na tabela *BodyValues*: *bodyTemperature* e *oxygenSaturion*. Eletrocardiograma é uma nova perspectiva de mostrar o batimento cardíaco, mas devido a sua complexidade e enorme informação que se pode tirar [115] é considerado uma nova categoria.

Concluindo, o impacto da adição de uma nova fonte de dados é mínimo, provocando algumas alterações ao modelo de dados, sendo que grande parte dos dados já estavam representados no modelo de dados.

6.3.2 Controlador - *SDictionary*

Acrescentar um novo controlador que comunique com uma nova fonte de dados é obrigatório em qualquer caso, ao adicionar um novo controlador o bom funcionamento do sistema não é afetado devido à modularidade do sistema. Como por exemplo, quando um novo funcionário é introduzido numa empresa, o impacto que tem na empresa é mínimo. Ou seja, a empresa funciona normalmente, é necessário explicar ao novo colaborador como a empresa funciona. O tempo que se demora a explicar ao novo funcionário o funcionamento da empresa é (por analogia) o tempo necessário para alterar a base de dados (resultando numa nova base de dados mais rica), juntamente com o tempo necessário para desenvolver um novo controlador que comunique com a API e guarde os dados na base de dados (sendo que a maior parte do código de comunicação com base de dados é aproveitado dos outros controladores).

6.4 Possíveis soluções para reduzir o impacto de alterações na base de dados

Quando é necessário alterar o modelo de dados alguns problemas surgem, como por exemplo alterar uma tabela já existente. Para atacar esse problema existe várias alternativas sendo que nenhuma delas é perfeita, tendo os seus prós e contras. De acordo com o livro *SQL Antipatterns: Avoiding the Pitfalls of Database Programming* [116] existem várias alternativas, entre as quais se destacam:

- *Extra Columns*;
- *Entity-Attribute-Value (EAV)*;
- *Class Table Inheritance*;
- *NoSQL*.

Antes de começar a descrição de cada uma das possíveis soluções é importante perceber que o tipo de base de dados usado é do tipo SQL³.

A solução *Extra Columns* consiste em criar novas colunas⁴ que possam ter o valor de nulo (vazio) para que possam conter informação extra. Esta solução pode não ser 100% fiável, podendo levar a que as várias colunas extras criadas não sejam suficientes, sendo que seja necessário criar novas colunas, voltando ao problema novamente.

Entity-Attribute-Value (EAV) pressupõe que sejam criados dois campos que definam o nome do atributo juntamente com o valor da leitura. Esta solução faz sentido para ser usada na tabela *BodyValues*, porque neste caso existem vários atributos que tem apenas um valor (índice de massa corporal, peso, altura e massa gorda). O que pode acontecer é que nem sempre seja assim tão linear, ou seja, podem existir atributos que necessitam de vários valores.

Para resolver este problema surgiu outra solução semelhante à EAV denominada por: *Class Table Inheritance*. É um conceito semelhante ao conceito de herança usado no paradigma orientado a objetos⁵, consiste na criação de uma tabela mãe em que todas as outras dependem dela, usando uma relação um para um. Assim, poderia existir uma tabela *BodyValues* (tabela mãe) e quatro tabelas filho denominadas por *Weight*, *BMI*, *Height* e *Fat*, que estendiam os atributos da tabela mãe. Esta alternativa já contrariava o facto de conter apenas um valor (da solução EAV), fazendo com que as tabelas filho consigam definir um conjunto maior de atributos. O grande problema deste abordagem é que se o sistema cresce faz com que cria um complexidade grande, fazendo com que fique muito complicado manter a base de dados.

Com os vários estudos e soluções pensadas, foi possível chegar à conclusão de que para certos projetos o melhor seria não existir um esquema à priori, surgindo o conceito de base de dados *NoSQL*. Uma base de dados *NoSQL* não necessita que um esquema seja definido e pode ser de vários tipos: *document-based*, *key-value pairs*, *graph databases* ou *wide-column stores* [117].

Destas todas soluções, a melhor alternativa é usar uma base de dados *NoSQL*, mas como uma das restrições impostas pela empresa foi usar uma base de dados SQL fez com que esta opção não fosse considerada.

³SQL significa *Structured Query Language*, basicamente um base de dados SQL é uma base de dados relacional, em que as entidades são denominadas por tabelas que estão fortemente ligadas, ou seja, pressupõe que se conheçam antes a estrutura de cada tabela, sendo que alterações a tabelas possam ficar complicadas.

⁴Coluna é um atributo que descreve um tipo de informação, como por exemplo *weight* é uma coluna da tabela *bodyValues* (figura 6.5).

⁵É um paradigma que consiste num modelo de programação de software baseado na composição e interação entre diversas unidades chamadas de *objeto*.

6.5 Solução tecnológica

Na arquitetura (figura 6.9) é possível identificar quatro componentes principais:

- Utilizador (paciente).
- Aplicação.
- *Wearable*.
- Base de dados.

Na aplicação existe um componente que trata de todas as comunicações com a API(s), é identificado este componente devido ao facto da informação que é transmitida ser sensível (sendo necessário um componente focado, apenas, nessa tarefa), o componente pode ser interno ao externo à aplicação. No lado da base de dados, está presente um componente que trata apenas de todas as comunicações internas e externas à base de dados, mais uma vez por uma questão de segurança e modularidade.

O fluxo de comunicação é simples, o utilizador tem acesso a um conjunto funcionalidades disponível numa aplicação (que suporta tanto *iOs* como *Android*). Por sua vez, a aplicação consome API(s) fornecidas por uma (ou várias) das fontes (secção 5.2), por fim os dados são guardados numa base de dados.

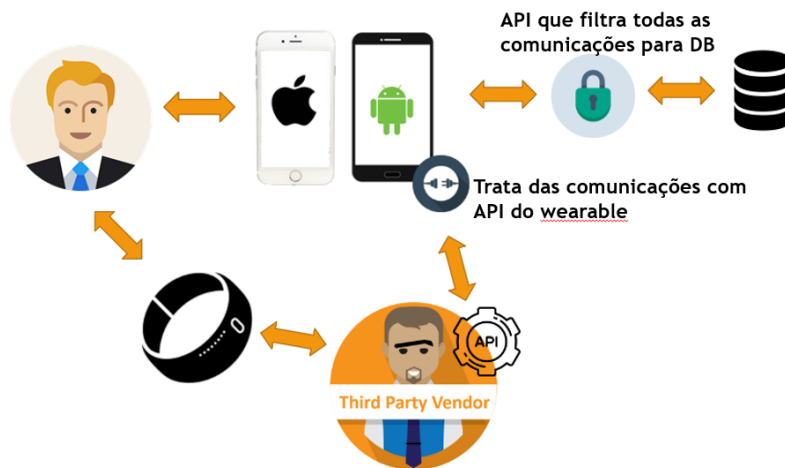


Figura 6.9: Arquitetura de alto nível

Capítulo 7

Metodologia de desenvolvimento

Este capítulo tem como objetivo dar uma visão geral das metodologias por detrás do projeto, juntamente com todas as ações consideradas de apoio ao desenvolvimento do mesmo. Portanto, o capítulo encontra-se dividido em 3 secções:

- **7.1:** descreve a metodologia usada no desenvolvimento do sistema de software.
- **7.2:** explica o planeamento do projeto.
- **7.3:** explica como foram realizadas as reuniões ao longo do projeto.

7.1 Metodologia de desenvolvimento de software

Antes de entrar na metodologia usado no desenvolvimento de software, é necessário referir que o período de desenvolvimento do sistema não é total à duração do projeto. Numa fase inicial do projeto, foi desenvolvido e pesquisado o estado de arte (primeiros 6 meses) e só na fase de desenvolvimento (segundos 6 meses) do projeto é que foi implementada a metodologia: *Agile*.

Agile é uma abordagem iterativa que consiste no desenvolvimento do software de uma forma incremental. Basicamente, permite dividir o desenvolvimento de software em iterações de duração curta (como por exemplo, 15 dias), havendo entregas do produto no fim de cada iteração. Esta metodologia assenta em vários pilares sendo os que fazem sentido realçar são:

- Satisfação do cliente desde inicio e entrega contínua de software de valor.
- Abertura à mudança constante dos requisitos desde do inicio até ao fim.
- Trabalho em conjunto com desenvolvedores e responsáveis pelo produto.
- Atenção contínua a um código de excelência e a um bom design e arquitetura.

Outro aspeto importante a referir nesta secção, é a forma como foram representados os requisitos funcionais. Os requisitos funcionais retratam as funcionalidades que o produto oferece ao utilizador, com por exemplo registo de utilizador, listar amigos, entre outros. Para definir os requisitos funcionais neste projeto foram usadas *user stories*.

User stories são uma forma simples e curta de descrever uma funcionalidade na perspetiva do utilizador (que irá usar a funcionalidade) e seguem a seguinte estrutura: "Como <tipo de utilizador>, quero <objetivo> para <motivo>". Foi escolhida esta forma de especificar requisitos funcionais devido ao facto de ser muito usado no processo de desenvolvimento *Agile* e, também, por serem escritas na perspetiva do utilizador, ajudando a perceber melhor o que o utilizador pretende ao usar a funcionalidade [118].

7.2 Planeamento

O planeamento foi dividido em duas partes, descrito nas figuras 7.1 e 7.2. Numa primeira fase, foi feita uma pesquisa detalhada para perceber o que já existe, para se conseguir entender o que se consegue acrescentar de novo a este conceito, resultando na escrita de um documento sobre o estado da arte. A partir de uma pesquisa inicial, foram definidos os temas principais a explorar no estado da arte (listagem dos wearables, FDA e EC, segurança dos dados de saúde, *Patient Generated Health Data*, entre outros temas presentes nos capítulos 2, 3 e 4), esta fase decorreu desde 18 de setembro até 31 de dezembro de 2017. A segunda fase foi constituída por três tarefas principais:

- A primeira tarefa consistiu na revisão do estado da arte e num contínuo levantamento dos requisitos.
- A segunda fase passou pelo desenvolvimento (usando a metodologia *Agile*) e a última semana (18 a 22 de junho) serviu para rever o documento da dissertação. É importante ainda referir que, a escrita do documento da dissertação é transversal a todo projeto e foi constantemente desenvolvida.

A fase de desenvolvimento que decorreu de 8 de março até 15 de junho incluiu as seguintes tarefas:

- 8 de março até 29 de março - estudo de cada fonte de dados (*Fitbit* e *Google Fit*, secção 6.2), com entregas (nos dias 16 e 29 de março) do modelo de dados de cada fonte;
- 2 de abril até 13 de abril - explorado o *React Native* (secção 8.1.1), com entregas do protótipo da aplicação móvel (nos dias 6 e 13 de abril);
- 16 de abril até 11 de maio - implementação dos componentes (base de dados, *FDictionary*, *GDictionary* e API, secção 8.3), que incluiu as seguintes sub-tarefas:
 - desenho do modelo unificado dos dados: 16 de abril até 20 de abril (entrega no dia 20 de abril);
 - implementação do modelo na base de dados: 23 de abril até 27 de abril (entrega no dia 27 de abril);
 - desenvolvimento dos conectores de comunicação com as fontes de dados (*FDictionary* e *GDictionary*): 30 de abril até 4 de maio (entrega no dia 4 de maio);

- desenvolvimento da API e ligação à aplicação móvel: 7 de maio até 11 de maio (entrega no dia 11 de Maio);
- 14 de maio - 15 de junho: melhoria do o projeto na preservativa da empresa e finalização da da escrita dissertação, com entregas à sexta-feira de 2 em 2 semanas. A duração de cada iteração passou a ser de 2 semanas, porque foi necessário começar a dedicar mais tempo a escrita da dissertação, sendo necessário mais tempo para entregar um produto com valor.

É importante mencionar, que foi possível aplicar a metodologia *Agile*, fazendo com que o projeto fosse crescendo como um todo e cada *iteração* teve duração de 1 semana. Na fase de desenvolvimento houve uma redefinição constante aos requisitos e à arquitetura do sistema, sempre que necessário.



Figura 7.1: Planeamento primeira fase

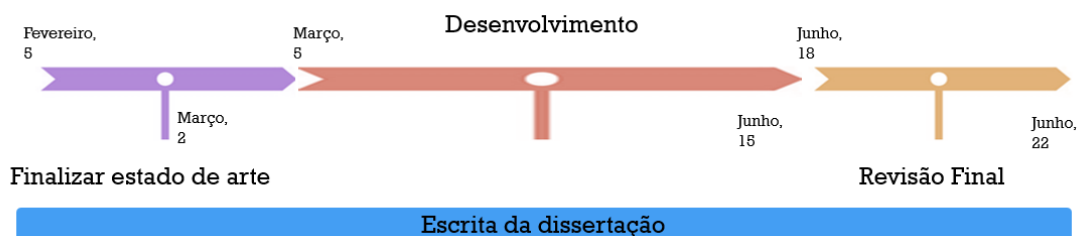


Figura 7.2: Planeamento segunda fase

7.3 Reuniões

Numa fase inicial, houve uma reunião introdutória ao projeto com o orientador da FEUP e o supervisor da empresa para definir e alinhar as expectativas da faculdade com as da empresa. De seguida, houve reuniões semanais com o orientador da FEUP e o supervisor da empresa (separadamente). Estas reuniões serviram para que os outros intervenientes do projeto estivessem em contacto constante com o desenvolvimento do projeto, e também para receber *feedback* valioso e ajudar em decisões de mais alto nível.

Capítulo 8

Desenvolvimento da solução

Neste capítulo é explicado, de um modo mais detalhado, a implementação dos componentes do projeto. Numa fase inicial, são abordadas as *frameworks* e linguagens de programação usadas para desenvolver a solução (secção 8.1). De seguida, são detalhadas as fases da Engenharia de Software:

- Requisitos: levanta os vários requisitos do projeto (secção 8.2).
- Arquitetura: define a arquitetura (componentes e suas comunicações, na secção 8.3).
- *Design*: aborda aspetos de mais baixo nível, incluindo métodos HTTP usados para comunicar com os fornecedores, bem como o fluxo de comunicação de cada requisito (secção 8.4).
- Testes: aborda e justifica os testes feitos à solução (secção 8.5).
- Validação (secção 8.6): prova a veracidade da informação usada no sistema.

8.1 Linguagens de Programação e *Frameworks*

Esta secção tem como objetivo mostrar e explicar as *frameworks* e linguagens de programação que este projeto tem por base. Sendo assim, a secção está dividida em duas secções que abordam três tecnologias:

- Secção 8.1.1 - *React Native* [119]
- Secção 8.1.2 - *Node.js* [120] e *MySQL* [121].

8.1.1 *React Native*

React Native [119] foi a *framework* escolhida para desenvolver o protótipo da aplicação móvel. *React Native* é uma *framework* que permite desenvolver aplicações móveis (em *Android* e *iOs*) nativas através do uso de *Javascript* [122] e *React* [123]. Antes de entrar nas vantagens do *React Native*, é importante perceber dois conceitos: *Javascript* e *React*.

Javascript é uma linguagem de programação interpretada, que foi desenvolvida com o propósito de oferecer dinamismo a web-sites (correndo no lado do cliente). Mas no últimos anos sofreu uma mudança, passando a ser usado, também, no lado do servidor (como por exemplo através do *Node.js* [120]).

React é uma biblioteca implementada em *Javascript*, desenvolvida e mantida pelo *Facebook* [124]. *React* tem como objetivo mudar um pouco o paradigma de desenvolvimento *front-end* de páginas web. Permite de um modo simples melhorar o desempenho e escalabilidade das páginas web, reutilizando componentes (componente pode ser qualquer parte de uma página web que possa ser reutilizado, como por exemplo um botão, uma entrada de texto, entre outros).

Relativamente ao *React Native*, existem um conjunto de vantagens que o realçam em comparação com a concorrência [125]:

- *Native Code* - Em comparação com outras *frameworks*, o *React Native* permite desenvolver em *Javascript* mas gera o código nativo, sendo possível obter um desempenho alto.
- Compilação - Ao desenvolver aplicações móveis um dos problemas é a compilação que é um processo moroso, para isso o *React Native* oferece uma alternativa. Através de uma ferramenta denominada por *Expo* [126], é possível correr a aplicação num servidor local (computador no qual se está a desenvolver) através de um URL, permitindo assim simular de uma forma simples a solução.
- *Maximum code reuse* - Com uma abordagem orientada a componentes a reutilização do código aumenta, permitindo um desenvolvimento mais rápido.

Para além das suas vantagens, a escolha recaiu no *React Native*, devido ao facto de ser uma *framework* nova que a empresa ainda não tinha usado e pretendia explorar.

8.1.2 *Node.js* e *MySQL*

Como já foi abordado na secção 6.5, o sistema para além de incluir uma aplicação móvel também contém uma camada que comunica com a base de dados e trata das comunicações com os fornecedores. De um ponto de vista de componentes (secção 8.3) inclui os componentes *FDictionary*, *GDictionary* e *API*. Sendo assim, a linguagem escolhida para desenvolver estes componentes foi o *Node.js* [120].

Node.js é uma *framework* que usa *Javascript* e destaca-se pelo as seguintes características [127]:

- *Event-driven*.
- *Non-blocking I/O model*.
- *NPM* [128].

Event-driven é um tipo de arquitetura orientada a eventos. Como a figura 8.1 mostra, existe apenas um *thread*¹ em que todos os eventos são registados, através do uso de *callbacks*² cada evento tem uma ação de resposta. O *event driven* é uma característica do *Node.js* e em conjunto com o *non-blocking I/O model* cria um sistema com um desempenho alto. *Non-blocking I/O model* define que um evento não deve bloquear o outro, ou seja, tem por base operações assíncronas. Outro aspeto em que esta ferramenta se destaca é a vasta oferta de *packages*³ que podem ser utilizadas através do *NPM*, oferecendo uma forma simples de obter um conjunto extra de serviços. Foi escolhido o *Node.js* por ser uma *framework* leve e capaz de permitir montar um sistema de uma forma rápida e eficaz.

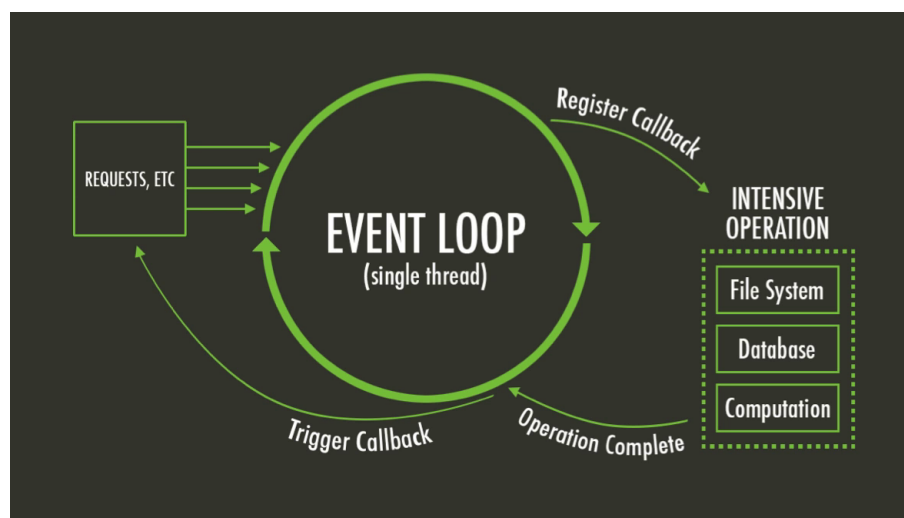


Figura 8.1: Arquitetura do *Node.js*

Para desenvolver a base de dados foi usado o sistema de gestão *MySQL* [121] que usa a linguagem SQL. Contem várias características entre as quais se destacam [129]:

- Excelente desempenho.
- Sem custos associados.
- Controlo transacional.
- Fácil configuração.

A escolha recaiu no *MySQL*, essencialmente, porque é uma ferramenta grátis, de simples configuração (assim como o *Node.js*) e por ser uma base de dados SQL (restrição imposta pela empresa).

¹É uma forma em que um processo de um sistema operativo se pode dividir de forma a ser executado concorrentemente, para obter um melhor desempenho.

²É uma função que é passada como parâmetro para ser executada no fim de outra função.

³Serviço extra que pode ser usado, como por exemplo um serviço que comunica com uma base de dados.

8.2 Requisitos

Esta secção tem como objetivo listar e explicar em mais detalhe os requisitos do projeto, consequentemente encontra-se dividida em duas secções:

- Secção 8.2.1 - lista os requisitos funcionais.
- Secção 8.2.2 - lista os requisitos não funcionais e encaixa-os nas categorias do *FURPS+*.

8.2.1 Funcionais

As funcionalidades da aplicação desenvolvidas nesta prova de conceito são as seguintes:

- *REQ-FUNC1* - Como utilizador, quero ter acesso à minha informação bem como aos wearables que possuo, para conseguir ter uma visão geral dos meus dados principais e dos aparelhos que leem a minha informação vital.
- *REQ-FUNC2* - Como utilizador, quero ter acesso ao meu peso e percentagem de massa gorda, para ter uma perspetiva geral da minha evolução.
- *REQ-FUNC3* - Como utilizador, quero ter acesso ao meu histórico de batimento cardíaco, para poder visualizar um valor vital importante.
- *REQ-FUNC4* - Como utilizador, quero ter acesso ao meu histórico de exercícios físicos, para perceber a minha evolução a nível físico ao longo do tempo.
- *REQ-FUNC5* - Como utilizador, quero ter acesso ao meu histórico de dados relativos à minha qualidade de sono, para verificar se tenho dormido as horas necessárias.

Outros requisitos:

- *REQ-FUNC6* - Como utilizador, quero conseguir apagar a minha informação a qualquer momento.
- *REQ-FUNC7* - Como utilizador, quero conseguir exportar a minha informação para as minhas outras fontes de dados.

Para terminar, é importante referir que os requisitos *REQ-FUNC6* e *REQ-FUNC7*, não foram implementados nesta prova de conceito, porque foi definido que o objetivo principal é provar que é possível importar dados de fontes diferentes e guardar essa informação num modelo final unificado.

8.2.2 Não-Funcionais

Um requisito funcional é equivalente a uma funcionalidade do sistema (secção 7.1). Por outro lado, um requisito não-funcional é uma limitação ou restrição nos recursos ou no desenho do

sistema [130], serve para descrever o quão bem o sistema deve fazer o que faz (permite definir a qualidade do sistema).

Uma técnica que permite agrupar requisitos não-funcionais é conhecida pelo acrónimo de *FURPS+* [131]. Foi escolhida esta técnica por ser muito usada e por funcionar com uma regra simples de agrupar os requisitos, sem estabelecer uma estrutura muito rigorosa, ou seja, oferecendo liberdade de escolha.

FURPS+ engloba cinco características principais:

- **Funcionalidade:** Capacidade, reusabilidade e segurança;
- **Usabilidade:** Estética e consistência na interface do utilizador;
- **Confiabilidade** (em inglês *Reliability*): Disponibilidade, precisão dos cálculos do sistema e a capacidade de recuperação do sistema em caso de falha;
- **Desempenho** (em inglês *Performance*): *Throughput*⁴, tempo de resposta, tempo de recuperação e tempo de inicialização;
- **Manutenção** (em inglês *Supportability*): Testabilidade, adaptabilidade, capacidade de manutenção, compatibilidade, configuração, escalabilidade e localização.

O sinal + descreve o resto dos requisitos:

- **Requisitos de desenho** - Especifica ou restringe opções para desenhar um sistema;
- **Requisitos de implementação** - Especifica ou restringe o código ou a construção do sistema (padrões, linguagens de implementação e limites de recursos);
- **Requisitos de interface** - Especifica sistemas ou componentes externos com os quais o sistema tem de interagir e/ou restringe formatos de comunicações entre o sistema e componentes externos;
- **Requisitos físicos** - Especifica restrições físicas impostas pelo hardware.

Do ponto de vista de requisitos não-funcionais foram identificados os seguintes requisitos:

- *REQ-N-FUNC1* - Não se pode guardar credencias de acesso a outras plataformas no lado da aplicação móvel nem da base de dados.
- *REQ-N-FUNC2* - Para ter acesso aos dados do utilizador em cada plataforma (*Google Fit* e *Fitbit*) é obrigatório usar o protocolo *O-Auth 2.0*.
- *REQ-N-FUNC3* - Na API da *Google Fit* não é possível exportar dados, com o objetivo de estes serem visíveis ao utilizador nos produtos da *Google* (aplicação *Google Fit* e versão web).

⁴Quantidade de dados transferidos de um ponto para outro ou quantidade de dados processados em um determinado intervalo de tempo.

O requisito *REQ-N-FUNC2* encaixa-se no grupo de *restrições de interface* do FRUPS+. *REQ-N-FUNC3* também é considerada uma restrição, mas neste caso de *implementação*. Por fim, o requisito *REQ-N-FUNC1* encaixa-se na secção de *funcionalidade* devido ao facto de ser tratar de um requisito de segurança.

8.3 Arquitetura

Nesta secção é mostrado os principais componentes que fazem parte do projeto, bem como o processo de comunicação entre eles. O objetivo é dar uma perspetiva da arquitetura envolvente do sistema.

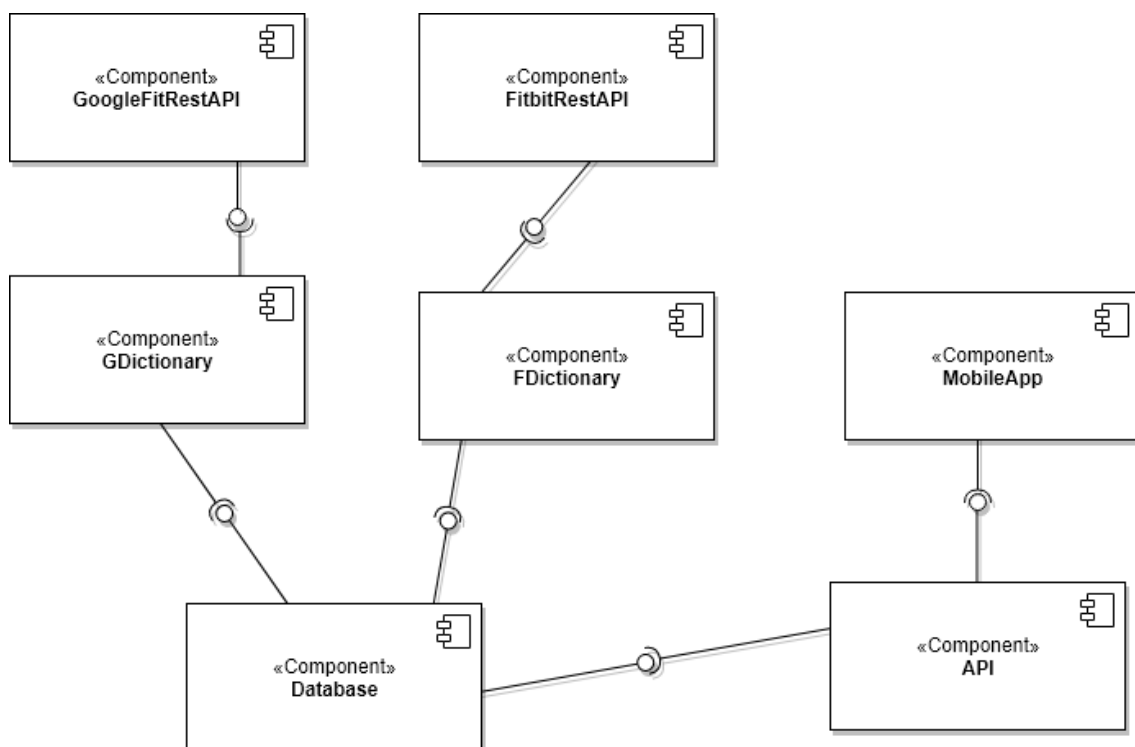


Figura 8.2: Diagrama de componentes

Como é possível perceber na imagem 8.2 existem sete componentes principais na aplicação:

- *GoogleFitRestAPI* - Representa a REST API que fornece os dados dos utilizadores da *Google Fit*.
- *FitbitRestAPI* - Representa a REST API que fornece os dados dos utilizadores da *Fitbit*.
- *GDictionary* - Trata de importar/exportar os dados da *Google Fit* no modelo de dados do projeto.

- *FDictionary* - Trata de importar/exportar os dados da *Fitbit* no modelo de dados do projeto.
- *Database* - Base de dados onde são armazenados os dados.
- *API* - API que consome os dados da base de dados e disponibiliza uma interface de comunicação.
- *MobileApp* - Aplicação móvel que mostra os dados ao utilizador.

O fluxo de informação é o seguinte: *GoogleFitRestAPI* fornece os dados ao *GDictionary* que por sua vez vai guardar os dados na base de dados, convertendo-os no modelo unificado de dados (o mesmo acontece com *FitbitRestAPI* e *FDictionary*). Numa segunda fase a *API* vai consumir os dados da base de dados e vai disponibiliza-los à *MobileApp*. É importante referir que o *GDictionary* e *FDictionary*, futuramente, também vão ter a responsabilidade de exportar os dados para, respetivamente, a *Google Fit* e *Fitbit*.

8.4 Design

O que distingue a etapa de *Design* da de Arquitetura (secção 8.3) é granularidade, ou seja, a etapa de *Design* tem uma granularidade mais baixa, em que o processo é mais detalhado. Sendo assim, esta secção está dividida em três partes:

- 8.4.1: estrutura interna da aplicação móvel.
- 8.4.2: detalha a comunicação entre o API e a aplicação móvel.
- 8.4.3: detalha a comunicação entre o sistema (componentes *FDictionary* e *GDictionary*) e as fontes de dados (componentes *FitbitRestAPI* e *GoogleFitRestAPI*).

8.4.1 Componentes da aplicação móvel

De um ponto de vista de componentes da aplicação móvel (figura 8.3) existem quatro componentes principais: *App*, *Screens*, *Handlers* e *API*.

O componente *App* é o componente principal que serve como ponto de arranque da aplicação, este componente vai usar os vários ecrãs disponíveis no componente *Screens*, que por sua vez usará os serviços disponíveis no componente *Handlers* para comunicar com a *API* que fornece os dados.

8.4.2 Fluxo comunicação entre API e Aplicação Móvel

Antes de iniciar a explicação de como é feito o fluxo de comunicação entre a API que consome a base de dados e a aplicação móvel, é importante perceber que para explicar de uma forma visual este fluxo foram usados diagramas de sequência⁵.

⁵Tipo de diagrama da linguagem *Unified Modeling Language* (UML) para representar a comunicação entre componentes de um sistema.

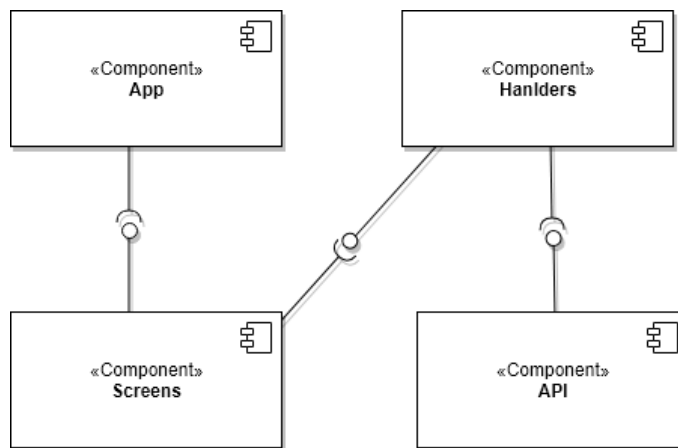


Figura 8.3: Diagrama de componentes da aplicação móvel

A estrutura principal de cada diagrama de sequência é idêntica o que muda são os métodos HTTP da API que são chamados e como é mostrada a informação ao utilizador.

Nas sub-seções seguintes (8.4.2.1, 8.4.2.2, 8.4.2.3, 8.4.2.4 e 8.4.2.5) são apresentados os diagramas de sequência que retratam o fluxo da informação de cada requisito funcional implementado na solução.

8.4.2.1 REQ-FUNC1

Neste requisito o principal objetivo é mostrar dados do utilizador considerados genéricos, como o nome, idade e género, juntamente com uma listagem dos wearables, como a figura 8.4 mostra.

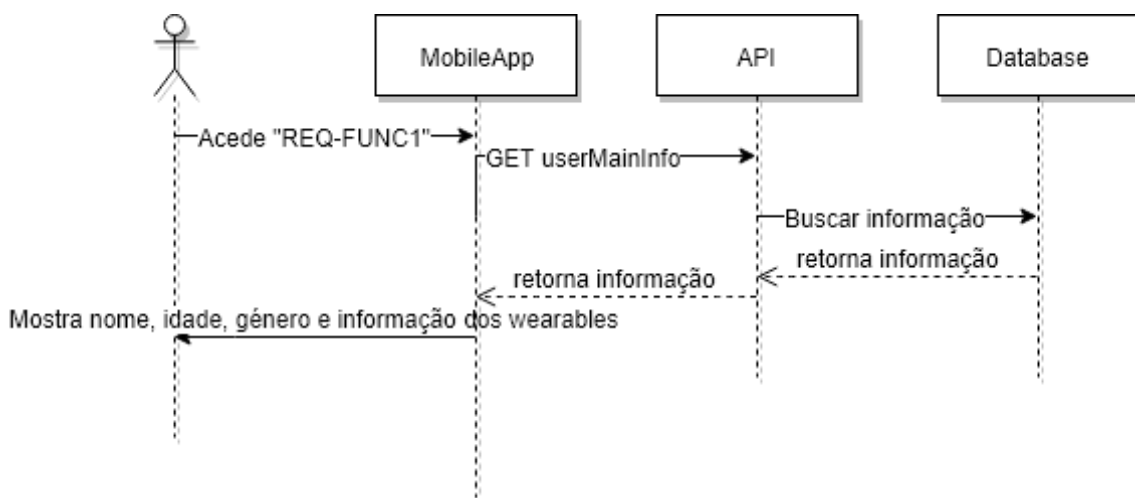


Figura 8.4: REQ-FUNC1 Diagrama de Sequência

8.4.2.2 REQ-FUNC2

No segundo requisito (figura 8.5) o principal objetivo é mostrar o histórico dos valores do peso e massa gorda num gráfico.

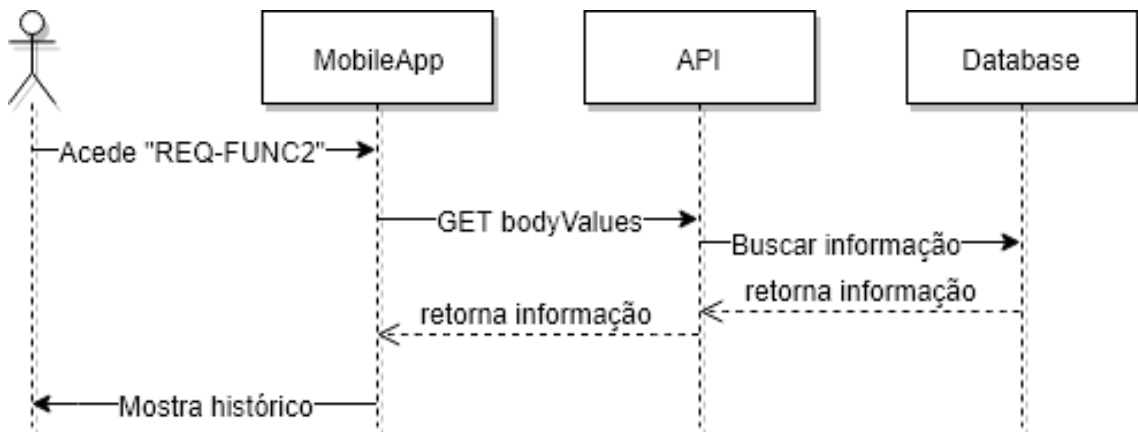


Figura 8.5: REQ-FUNC2 Diagrama de Sequência

8.4.2.3 REQ-FUNC3

Neste requisito é mostrado um histórico dos valores do batimento cardíaco do utilizador (como a imagem 8.6 detalha).

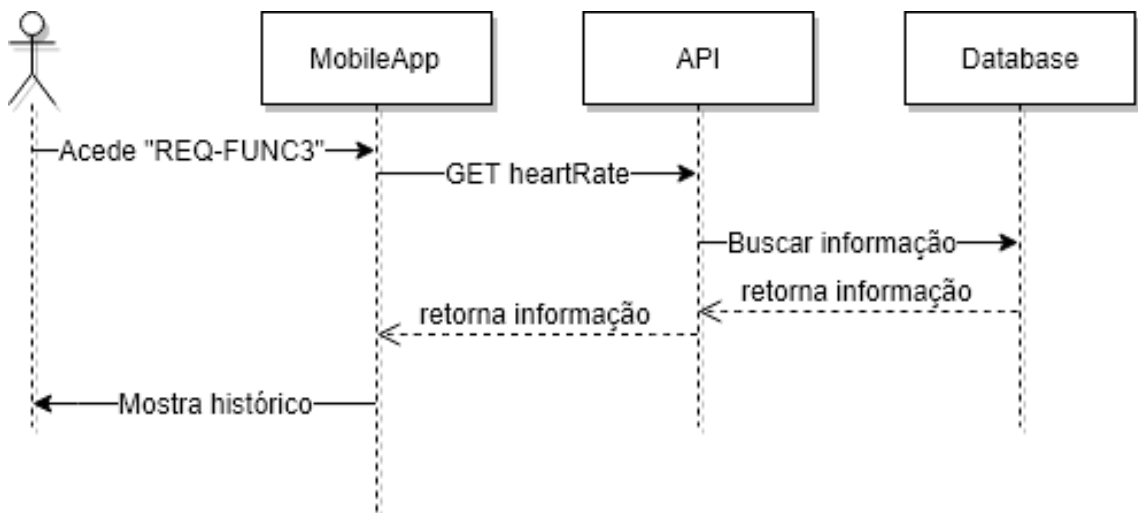


Figura 8.6: REQ-FUNC3 Diagrama de Sequência

8.4.2.4 REQ-FUNC4

O requisito *REQ-FUNC4* (figura 8.7) é idêntico ao *REQ-FUNC3* a diferença está na informação mostrada. No *REQ-FUNC4* a informação a ser mostrada é o histórico dos exercícios feitos pelo utilizador com os valores-resultado dos exercícios: distância, passos e calorias consumidas.

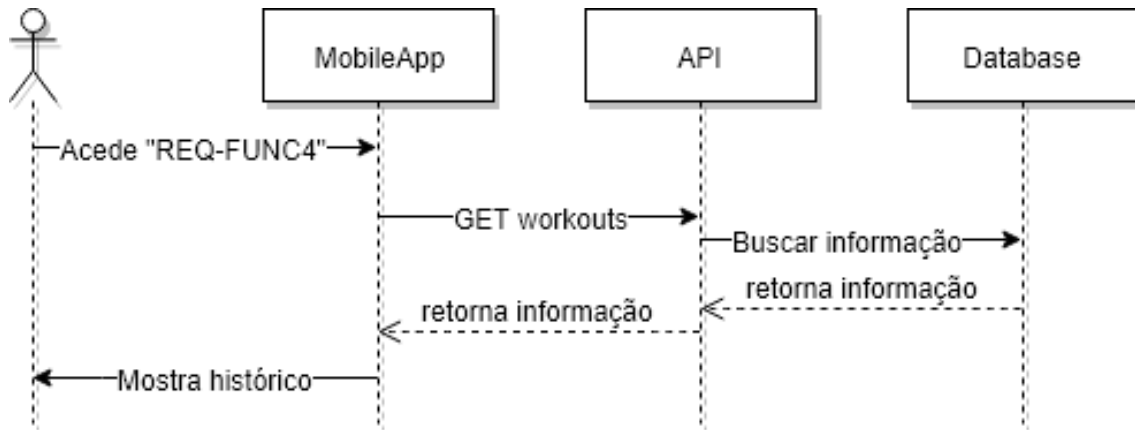


Figura 8.7: *REQ-FUNC4* Diagrama de Sequência

8.4.2.5 REQ-FUNC5

No requisito *REQ-FUNC5* (figura 8.8) é mostrado um histórico das horas dormidas.

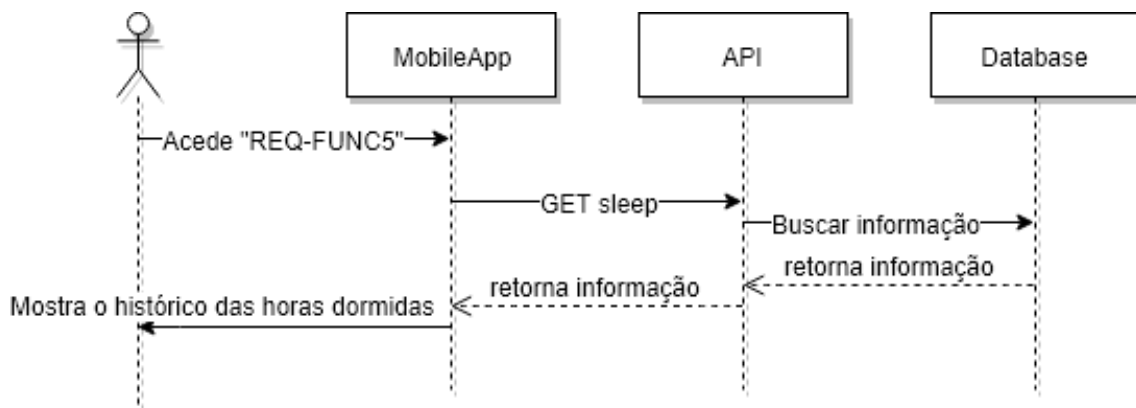


Figura 8.8: *REQ-FUNC5* Diagrama de Sequência

8.4.3 Fluxo comunicação entre as fontes de dados e o sistema

Na secção anterior (8.4.2), é explicado em mais detalhe a comunicação entre a API e a aplicação móvel, nesta secção o objetivo é semelhante, mas o foco principal é a comunicação entre os componentes *FDictionary* e *GDictonary* e as APIs da *Fitbit* e *Google Fit*, respetivamente. Estando esta secção dividida em outras duas secções:

- 8.4.3.1 - Métodos HTTP consumidos na REST API da *Fitbit* [89].
- 8.4.3.2 - Métodos HTTP consumidos na REST API da *Google Fit* [102].

8.4.3.1 *Fitbit*

Tendo em conta a tabela 8.1, é importante perceber que todos os métodos são do tipo GET e todos retornam o resultado em JSON⁶.

Relativamente ao requisito *REQ-FUNC1* existem dois métodos: o primeiro serve para retornar a informação principal do utilizador (dados do utilizador e configuração da plataforma) e o segundo serve para listar os *wearables* do utilizador. No requisito *REQ-FUNC2* são retornados os valores do peso, massa gorda e BMI num intervalo de tempo (*DataInicio* e *DataFim*). O mesmo se aplica aos requisitos *REQ-FUNC3* e *REQ-FUNC5*, só que neste caso é retornada informação relativa ao batimento cardíaco e sono, respetivamente. Por fim, o *REQ-FUNC4* recebe apenas uma data, que retorna as atividades desse dia, ou seja, neste caso tem que se fazer vários pedidos com as datas pretendidas. As datas devem estar no formato *AAAA-MM-DD*, em que *AAAA* é o ano, *MM* é o mês e *DD* é o dia.

Requisito	URL
<i>REQ-FUNC1</i>	<code>https://api.fitbit.com/1/user/-/profile.json</code>
<i>REQ-FUNC1</i>	<code>https://api.fitbit.com/1/user/-/devices.json</code>
<i>REQ-FUNC2</i>	<code>https://api.fitbit.com/1/user/-/body/log/weight/date/DataInicio/DataFim.json</code>
<i>REQ-FUNC3</i>	<code>https://api.fitbit.com/1/user/-/activities/heart/date/DataInicio/DataFim.json</code>
<i>REQ-FUNC4</i>	<code>https://api.fitbit.com/1/user/-/activities/date/Data.json</code>
<i>REQ-FUNC5</i>	<code>https://api.fitbit.com/1.2/user/-/sleep/date/DataInicio/DataFim.json</code>

Tabela 8.1: Métodos HTTP da fonte *Fitbit*

8.4.3.2 *Google Fit*

O mesmo que se aplicou aos métodos da API anterior (8.4.3.1), aplica-se a este API: os métodos são GET e retornam JSON. À exceção da primeira linha da tabela 8.2, os URLs estão incompletos faltando juntar ao início `https://www.googleapis.com/`. Relativamente aos parâmetros das datas, tem uma representação diferente (à exceção da do primeiro método do requisito *REQ-FUNC4*, em que representação é igual a da *Fitbit*, secção 8.4.3.1), em que é usada a representação denominada por *Unix Timestamp*, que consiste na diferença em milissegundos entre a data 1 de janeiro de 1970 e a data que se pretende.

Através da análise da tabela 8.2 consegue-se perceber que para todos os métodos, à exceção da primeira e quarta linhas da tabela, falta especificar o tipo de dados dos pedidos, que são os seguintes:

- *REQ-FUNC2* - `raw:com.google.weight:com.google.android.apps.fitness:user_input;`

⁶*JavaScript Object Notation* é um formato bastante usado na partilha de informação, definindo uma forma simples de representar a informação.

- *REQ-FUNC3* - *com.google.heart_rate.bpm*;
- *REQ-FUNC4* - *raw:com.google.distance.delta:com.google.android.apps.fitness:user_input*;
- *REQ-FUNC4* - *raw:com.google.step_count.delta:com.google.android.apps.fitness:user_input*;
- *REQ-FUNC4* - *raw:com.google.calories.expended:com.google.android.apps.fitness:user_input*.

Com os pedidos HTTP explicados falta perceber o que cada pedido faz:

- *REQ-FUNC1* - Retorna a informação do utilizador.
- *REQ-FUNC2* - Lista o peso do utilizador.
- *REQ-FUNC3* - Lista os valores de batimento cardíaco do utilizador.
- *REQ-FUNC4* - Lista sessões do utilizador.
- *REQ-FUNC4* - Lista a distância.
- *REQ-FUNC4* - Lista a contagem dos passos.
- *REQ-FUNC4* - Lista calorias queimadas.

Para terminar é importante perceber que, como já explicado na secção 6.1.1, a abordagem da *Google* é orientada a tipo de dados, ou seja, cada pedido é referente a um tipo de dados. Consequentemente, o requisito *REQ-FUNC4* envolve mostrar a informação dos exercícios físicos (sessões). Para isso, é necessário fazer um pedido para obter as sessões todas e só depois fazer um pedido a cada resultado da sessão com o respetivo tipo de dados (distância, passos e calorias), juntamente com a data de início e fim de cada sessão.

Requisito	URL
<i>REQ-FUNC1</i>	<i>https://www.googleapis.com/oauth2/v1/userinfo?alt=json</i>
<i>REQ-FUNC2</i>	<i>fitness/v1/users/me/dataSources/tipoDeDados/datasets/DataInicio-DataFim</i>
<i>REQ-FUNC3</i>	<i>fitness/v1/users/me/dataSources/tipoDeDadosDataInicio-DataFim</i>
<i>REQ-FUNC4</i>	<i>fitness/v1/users/me/sessions?startTime=DataInicio&endTime=DataFim</i>
<i>REQ-FUNC4</i>	<i>fitness/v1/users/me/dataSources/tipoDeDados/datasets/DataInicio-DataFim</i>
<i>REQ-FUNC4</i>	<i>fitness/v1/users/me/dataSources/tipoDeDados/datasets/DataInicio-DataFim</i>
<i>REQ-FUNC4</i>	<i>fitness/v1/users/me/dataSources/tipoDeDados/datasets/DataInicio-DataFim</i>

Tabela 8.2: Métodos HTTP da fonte *Google Fit*

8.5 Testes

Para provar se os dados foram realmente gravados na base de dados foram realizados testes unitários à API para verificar se estava a ler os valores corretamente. A lista de testes é a seguinte:

- **User Data:** verifica se o utilizador foi criado com sucesso.

- **User Fitbit Config:** verifica se a configuração foi corretamente criada para a *Fitbit*.
- **User Google Fit Config:** verifica se a configuração foi corretamente criada para a *Google Fit*.
- **Body Values - Fitbit:** verifica se os valores corporais foram criados e verifica se a fonte registada é da *Fitbit*.
- **Body Values - Google Fit:** verifica se os valores corporais foram criados e verifica se a fonte registada é da *Google Fit*.
- **Activity - Fitbit:** verifica se os exercícios foram criados para a fonte da *Fitbit*.
- **Activity - Google Fit:** verifica se os exercícios foram criados para a fonte da *Google Fit*.
- **Heart Rate:** verifica se os dados referentes ao batimento cardíaco foram criados (apenas para a *Fitbit*).
- **Heart Rate Zone:** verifica se as zonas de batimento cardíaco da *Fitbit* foram criadas.
- **Sleep:** verifica se os dados principais do sono da *Fitbit* foram criados.
- **Sleep Data:** verifica se o sumário dos dados de sono do utilizador foram criados (apenas para *Fitbit*).
- **Sleep Type of Reading:** verifica se as várias leituras que compõe os dados de sono do utilizador foram criadas (apenas para *Fitbit*).

Com os testes necessários feitos ao componente *API* é possível provar que o restante sistema funciona, ou seja, se os dados chegam à base de dados convertidos no modelo unificado e a *API* fornece os dados de forma correto é a prova necessária que o sistema funciona. Para complementar a descrição dos testes, está presente nos anexos o resultado da execução dos mesmos (B.1).

8.6 Validação

Para provar o bom funcionamento da solução é preciso perceber o fluxo e ordem das operações de manipulação dos dados. Como já foi abordado na secção 8.3, existem cinco componentes desenvolvidos no projeto:

- *FDictionary* e *GDictionary*: comunicam com as fontes de dados (respetivamente *Fitbit* e *Google Fit*) e guardam os dados no modelo unificado;
- *API*: fornece os dados guardados na base de dados;
- Base de dados: guarda os dados;
- Aplicação Móvel: mostra a informação ao utilizador.

```

PS C:\Users\j3080\Documents\GIT\FGHD-FDictionary\pghd-fdictionary> node -\server.js
ENTRY 1 -> USER INFO
ENTRY 2 -> USER CONFIG
ENTRY 3 -> DEVICE
ENTRY 4 -> BODY VALUES
ENTRY 5 -> HEART RATE
ENTRY 6 -> ACTIVITY
ENTRY 7 -> SLEEP
connected to mysql
1
ENTRY 3
WRITE USER INFO FROM FITBIT
USER MAIN INFO DONE!
2
ENTRY 2
WRITE USER CONFIG FROM FITBIT
CONFIG INFO DONE!
3
ENTRY 3
WRITE DEVICES
DEVICES DONE!
4
ENTRY 4
WRITE BODY VALUES
BODY VALUES DONE!
5
ENTRY 5
WRITE HEART RATE VALUES
HEART RATE DONE!
6
ENTRY 6
WRITE ACTIVITY VALUES
WRITE ACTIVITY VALUES
ACTIVITY 2018-04-10 DONE!
ACTIVITY 2018-04-09 DONE!
7
ENTRY 7
WRITE SLEEP VALUES
SLEEP DONE!

```

Figura 8.9: Execução do *script* do componente *GDictionary*

```

PS C:\Users\j3080\Documents\GIT\FGHD-FDictionary\pghd-fdictionary> node -\server.js
ENTRY 1 -> USER INFO
ENTRY 2 -> USER CONFIG
ENTRY 3 -> DEVICE
ENTRY 4 -> BODY VALUES
ENTRY 5 -> HEART RATE (NO DATA)
ENTRY 6 -> SLEEP (NO DATA)
ENTRY 7 -> SLEEP
ENTRY 8 -> SLEEP
ENTRY 9 -> CALORIES
connected to mysql
ENTRY 2
USER CONFIG INFO INSERTED.
ENTRY 3
NO GOOGLE FIT ENTRY POINT FOR DEVICES INFO 1
ENTRY 4
WEIGHT INSERTED.
ENTRY 5
NO HEART RATE DATA ON GOOGLE FIT 5
ENTRY 6
ACTIVITY 1 INSERTED.
ACTIVITY 2 INSERTED.
ENTRY 7
NO SLEEP DATA ON GOOGLE FIT
ENTRY 8
DISTANCE 1 INSERTED.
DISTANCE 2 INSERTED.
ENTRY 9
STEPS 1 INSERTED.
STEPS 2 INSERTED.
ENTRY 10
CALORIES 1 INSERTED.
CALORIES 2 INSERTED.

```

Figura 8.10: Execução do *script* do componente

Com os componentes identificados, só falta perceber qual é o fluxo da informação. Numa primeira fase é necessário comunicar com os fornecedores (*Fitbit* e *Google Fit*) e guardar os dados na base de dados. De seguida, é necessário comunicar com a base de dados e mostrar os dados ao utilizador. Sendo assim, esta secção está dividida nas seguintes secções:

- 8.6.1: mostra a execução dos conectores com as APIs das fontes de dados.
- 8.6.2: mostra o resultado final dos dados guardados na base de dados.
- 8.6.3: mostra o resultado final da aplicação móvel com os dados da base de dados.
- 8.6.4: compara os resultados das base de dados com os dados de teste de cada plataforma.

8.6.1 Execução dos componentes *FDictionary* e *GDictionary*

Como já abordado na secção 8.1.2, estes componentes foram desenvolvidos com *Node.js*. Para isso foram desenvolvidos dois *scripts* que comunicavam com as APIs das fontes de dados, e com a informação de cada fonte mapeiam os dados no modelo unificado, através de chamadas à base de dados.

Pelas figuras 8.9 e 8.10 é possível perceber que cada *script* tem as várias opções disponíveis para cada fonte de dados, sendo que é necessário realçar que para a *Google Fit* não existem as opções de guardar dados de batimento cardíaco, devido ao facto de ser necessário usar um *wearable* para ter tal informação, e também não fornece dados de sono nem informação relativa aos aparelhos que o utilizador possui. Com a execução de cada componente os dados já se encontram armazenados na base de dados, logo o passo seguinte (secção 8.6.2) é provar que essa informação está realmente guardada na base de dados.

8.6.2 Resultado na base de dados

Para mostrar o resultado foi usada uma ferramenta que permite visualizar os dados na base de dados, denominada por *MySQL Workbench* [132]. Os dados apresentados nesta secção são os de

utilizador, os valores corporais e os exercícios físicos, os restantes resultados estão na secção D dos anexos.

UserID	firstName	lastName	dateOfBirth	gender	picture	user_cri	dt_cri	user_updt	dt_updt
1	João	Gomes	1995-07-28	MALE	https://static0.fitbit.com/images/profile/default...	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 8.11: Tabela *User*

Na tabela 8.11 é possível perceber os vários valores do utilizador, como o primeiro nome (*firstName*), último nome (*lastName*), data de nascimento (*dateOfBirth*), o sexo (*gender*) e URL da imagem do utilizador (*picture*). É importante referir que esta informação é do utilizador referente à *Fitbit*.

BodyValuesID	BodyValues_UserID	dateEntry	externalLogID	bmi	timeEntry	innerSource	weight	fat	sourceEntry	height	user_cri	dt_cri	user_updt	dt_updt
24	1	2018-04-09	1523318399000	24.66	23:59:59	API	79	14.5	FITBIT	1.79	NULL	NULL	NULL	NULL
25	1	2018-04-10	1523404799000	24.97	23:59:59	API	80	15	FITBIT	1.79	NULL	NULL	NULL	NULL
26	1	2018-04-11	1523491199000	24.81	23:59:59	API	79.5	15.5	FITBIT	1.79	NULL	NULL	NULL	NULL
27	1	2018-04-12	1523577599000	25.28	23:59:59	API	81	15.300000190734863	FITBIT	1.79	NULL	NULL	NULL	NULL
28	1	2018-04-13	1523663999000	24.34	23:59:59	API	78	13.699999809265137	FITBIT	1.79	NULL	NULL	NULL	NULL
29	1	2018-04-03		0		API-USER-INPUT	71	0	GOOGLE-FIT	1.79	NULL	NULL	NULL	NULL
30	1	2018-04-04		0		API-USER-INPUT	75	0	GOOGLE-FIT	1.79	NULL	NULL	NULL	NULL
31	1	2018-04-05		0		API-USER-INPUT	73	0	GOOGLE-FIT	1.79	NULL	NULL	NULL	NULL
32	1	2018-04-06		0		API-USER-INPUT	72.2	0	GOOGLE-FIT	1.79	NULL	NULL	NULL	NULL

Figura 8.12: Tabela *BodyValues*

Outra categoria a ser abordada nesta secção, é a dos valores corporais, devido ao facto de existir o valor de massa gorda, que a *Google Fit* não fornece, e assim é possível provar que essa diferença de informação está a ser gerida de forma correta. Pela figura 8.12 é possível perceber que os valores do índice de massa corporal (*BMI*) e massa gorda (*fat*) existem na fonte (*sourceEntry*) *Fitbit*, mas na *Google Fit* o valor é zero, ou seja, estes valores na *Fitbit* são fornecidos, mas na *Google Fit* não o são. O forma como estes valores são geridos é: verificar a fonte e colocar o valor a zero, sendo que toda essa lógica está centrada nos componentes *GDictionary* e *FDictionary*, não afetando a base de dados.

ActivityID	Activity_UserID	externalActivityID	externalParentID	calories	description	distance	durationMileseconds	hasStartTime	externalLogID	nameEntry	startTime	steps	sourceEntry	externalActivityParentName	externalLastModified
1	1	17151	90013	450	Walking less than 2 mph, strollino very slowly	7.8	7995000	1	13617327285	Walk	13:00	10404	FITBIT	Walk	2018-04-18T08:31:52.000Z
2	1	12030	90009	876	Running - 5 mph (12 min/mile)	9	2702000	1	13618108025	Run	18:02	7341	FITBIT	Run	2018-04-18T08:36:33.000Z
3	1	0	0	288	-	4	3000000	1	1522694800000	Walk	17:00	4567	GOOGLE-FIT	Walk	2018-04-22T18:36:32.775Z
4	1	0	0	385	-	5.6	4080000	1	1522765800000	Walk	15:33	3883	GOOGLE-FIT	Walk	2018-04-22T18:36:34.405Z

Figura 8.13: Tabela *Activity* parte 1

A importância por detrás da escolha da tabela *Activity* (figura 8.13) é perceber que valores como a distância (*distance*) são convertidos na mesma unidade (que neste caso é em quilómetros), mesmo sendo recebidos em formatos diferentes.

Concluindo, o próximo passo é mostrar que estes valores são visíveis na aplicação móvel, sendo consumidos através do componente *API* que comunica com a base de dados.

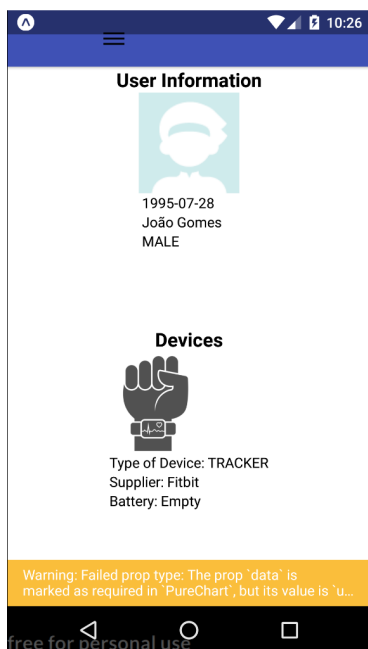
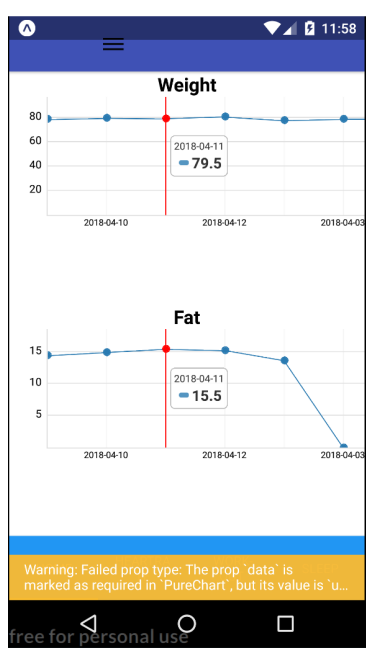
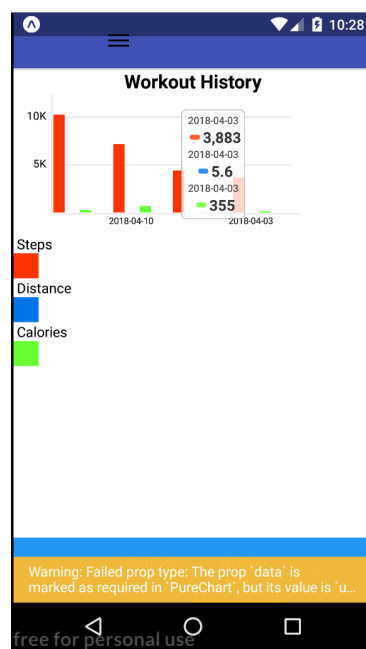
startDate	user_cri	dt_cri	user_updt	dt_updt
2018-04-09	NULL	NULL	NULL	NULL
2018-04-10	NULL	NULL	NULL	NULL
2018-04-02	NULL	NULL	NULL	NULL
2018-04-03	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Figura 8.14: Tabela *Activity* parte 2

8.6.3 Aplicação móvel

A abordagem nesta secção é igual a da secção anterior (8.6.2), sendo mostrados os dados das categorias *Utilizador*, *Valores Corporais* e *Exercício Físico*, mas neste caso tratados na aplicação móvel.

Como a figura 8.15 mostra, é possível verificar a veracidade dos dados já apresentados anteriormente na figura 8.11. O mesmo se aplica as figuras 8.16 e 8.12, em que é possível verificar as varias leituras com as respetivas datas, ainda é importante referenciar que nos valores de massa gorda (*Fat*) existem valores a zero que fazem referencia aos dados da *Google Fit*. Nas figuras 8.17, 8.13 e 8.14 o mesmo se verifica com os quatro exercícios físicos.

Figura 8.15: Ecrã da funcionalidade *REQ-FUNC1*Figura 8.16: Ecrã da funcionalidade *REQ-FUNC2*Figura 8.17: Ecrã da funcionalidade *REQ-FUNC4*

8.6.4 Fonte de dados

Para fechar este capítulo, só falta provar que os dados são oriundos das fonte de dados. Relativamente aos dados da *Fitbit*, é possível verificar o primeiro nome do utilizador (figura 8.18), o mesmo se verifica para os valores corporais (massa gorda, peso e as respetivas datas, presentes na figura 8.19), estes valores são iguais aos valores presentes nas tabelas (descritas nas figuras 8.11 e 8.12). Falta apenas analisar os valores relativos aos exercícios físicos. Os valores-resultado das atividades são as calorias, distância, passos e duração (figura 8.20), que também estão corretos em relação aos valores da tabela (presentes nas figuras 8.13 e 8.14).

De um ponto de vista dos dados da *Google Fit*, também é possível verificar os valores do peso (figura 8.23) e as duas atividades (figuras 8.21 e 8.22). Os restantes valores que são possíveis de visualizar nas plataformas estão presentes nos anexos na secção E.



Figura 8.18: Informação do utilizador na *Fitbit*











Week of Abr 9			79.5 kg
Abr 13	 	13.7% Fat	78 kg
Abr 12	 	15.3% Fat	81 kg
Abr 11	 	15.5% Fat	79.5 kg
Abr 10	 	15% Fat	80 kg
Abr 9	 	14.5% Fat	79 kg

Figura 8.19: Informação dos valores corporais na *Fitbit*






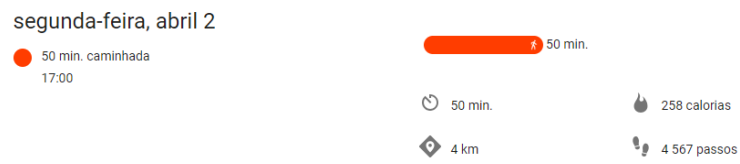
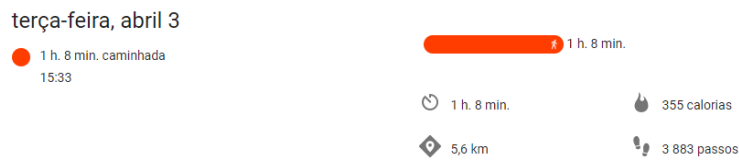
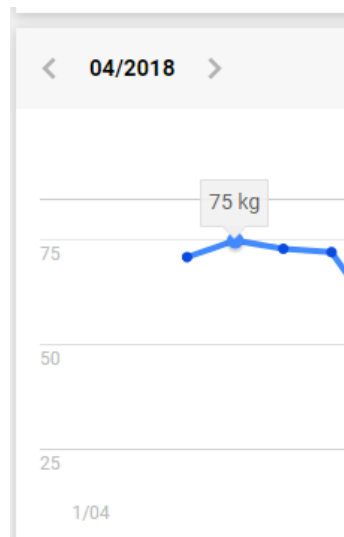
Activity History						
Date	 Activity	 Steps	 Distance	 Duration	 Calories	
10 Apr, 18:02	Run	7,341	9 kilometers	45:02	876 cal	View Details
9 Apr, 13:00	Walk	10,404	7.8 kilometers	2:13:15	450 cal	View Details

Figura 8.20: Informação dos exercícios físicos corporais na *Fitbit*

Figura 8.21: Exercício número 1 da *Google Fit*Figura 8.22: Exercício número 2 da *Google Fit*Figura 8.23: Informação dos valores corporais na *Google Fit*

Capítulo 9

Conclusão

Neste capítulo, abordam-se os objetivos atingidos bem como as contribuições feitas ao longo do projeto (secção 9.1). São igualmente elencadas as várias dificuldades encontradas (secção 9.2) e, por fim, as novas questões que foram surgindo ao longo do projeto e que ficam para trabalho futuro (secção 9.3).

9.1 Contribuições

Por ser um tema novo a nível da empresa e também por existirem poucas soluções que satisfaçam na totalidade o tema principal deste projeto (*Patient Generated Health Data*), o trabalho exploratório potencia-se vasto. Para além disso, juntando o tema dos *wearables*, que é um tema com soluções novas a surgirem de mês a mês, aumenta a esfera de informação a ser explorada, criando ainda mais desafios.

Este projeto pode ser visto como o projeto introdutório que junta duas áreas (*Patient Generated Health Data* e *wearables*) com muito potencial, permitindo abrir caminho a novos desafios (retratados no trabalho futuro 9.3). Sendo assim, é possível concluir que houve contribuições do ponto de vista industrial, conseguindo abrir um conjunto de questões que têm de ser abordadas para criar um produto fiável para o mercado. Conseguindo, por outro lado, criar uma solução capaz de provar que é possível juntar várias fontes de dados, que oferecem um conjunto de dados lidos e capturados por *wearables* de diferentes fornecedores. Também foi criada uma solução modular que permite a incorporação do papel do agente de saúde (através do desenvolvimento de outro componente equivalente ao componente da Aplicação Móvel, ver secção 8.3).

Por conseguinte, produziram-se contribuições nas seguintes áreas:

- *Patient Generated Health Data*: foi possível criar um sistema que armazene informação dos utentes, que facilmente pode ser estendido a uma outra plataforma para que os agentes de saúde possam analisar essa informação;
- *Wearables*: foi criado um sistema que junta informações de diferentes tipos de *wearables*, provando que é possível convergir num modelo comum, independente da fonte;

- Engenharia de Software: a criação de um sistema de software que envolve vários componentes e diferentes tipos de comunicação, produziu uma contribuição menor nesta área, mas mesmo assim, a registar.

De um modo geral, foi atingido o objetivo proposto no início da dissertação (secção 1.2) que era o desenvolvimento de uma solução que guardasse informação de *wearables* de fornecedores diferentes num modelo unificado, juntamente com a implementação de uma aplicação móvel que tratasse e apresentasse essa informação.

9.2 Dificuldades

De um ponto de vista das dificuldades consideradas graves e bloqueadores, estas não existiram, porque sempre foram ultrapassadas de um forma ou outra. A única dificuldade que é importante destacar, é o facto de não ter sido usado um *wearable*, em tempo de vida útil do projeto, para testar o protótipo, algo que foi ultrapassado através da introdução de dados de teste em cada plataforma.

9.3 Trabalho Futuro

Entre os novos problemas que foram surgindo a nível do desenvolvimento deste projeto e que podem ser considerados como trabalho futuro, destacam-se:

- sincronização da informação de fontes de dados diferentes;
- implementar uma base de dados que minimize (ou retire na totalidade) o impacto de alterações à mesma (possíveis soluções já abordados secção 6.4);
- implementar um sistema de acordo, na perfeição, com nova regulamentação (RGPD, possíveis soluções já abordadas na secção 5.4);
- implementar técnicas de segurança de informação de dados de saúde;
- exportar informação para as fontes de dados;
- adicionar o papel de agente de saúde;
- criar um sistema de utilizadores que se liga a base de dados de utentes da *Glinnt*;
- expandir o sistema para os outros contextos (secção 2.3);
- adicionar um maior conjunto de fontes de dados.

Anexo A

Modelo de dados das fontes

A.1 Modelo de dados da *Google Fit*

A.2 Modelo de dados da *Fitbit*

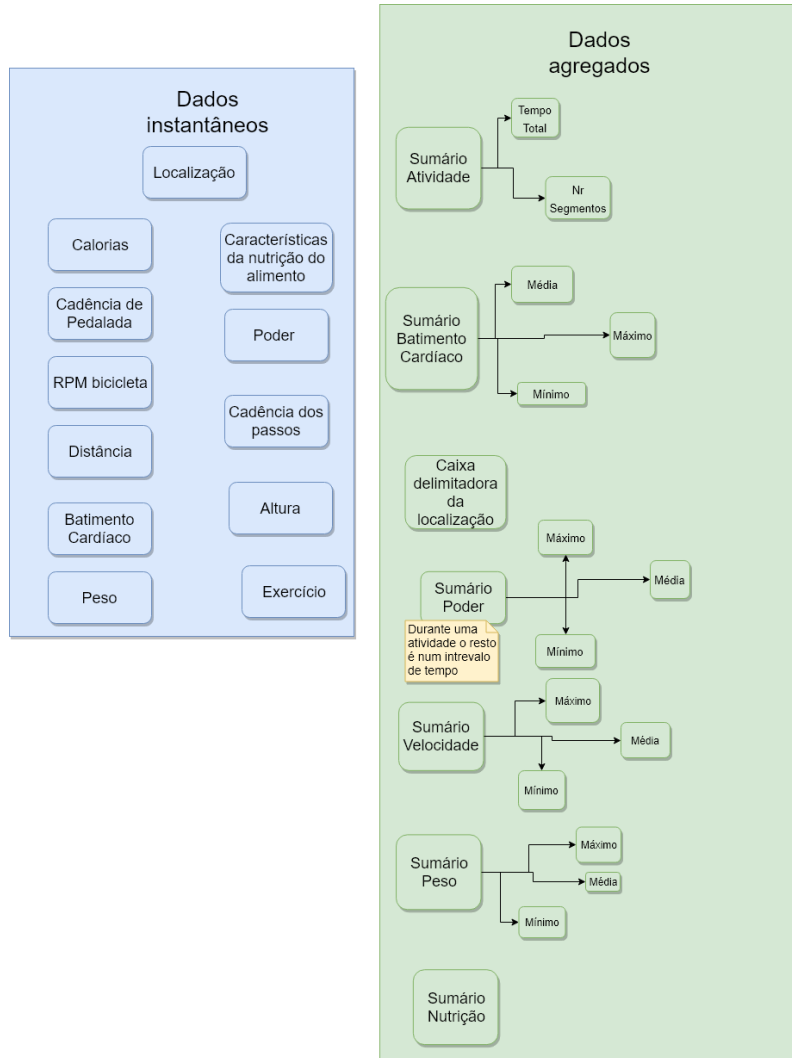


Figura A.1: Modelo de dados *Google Fit*

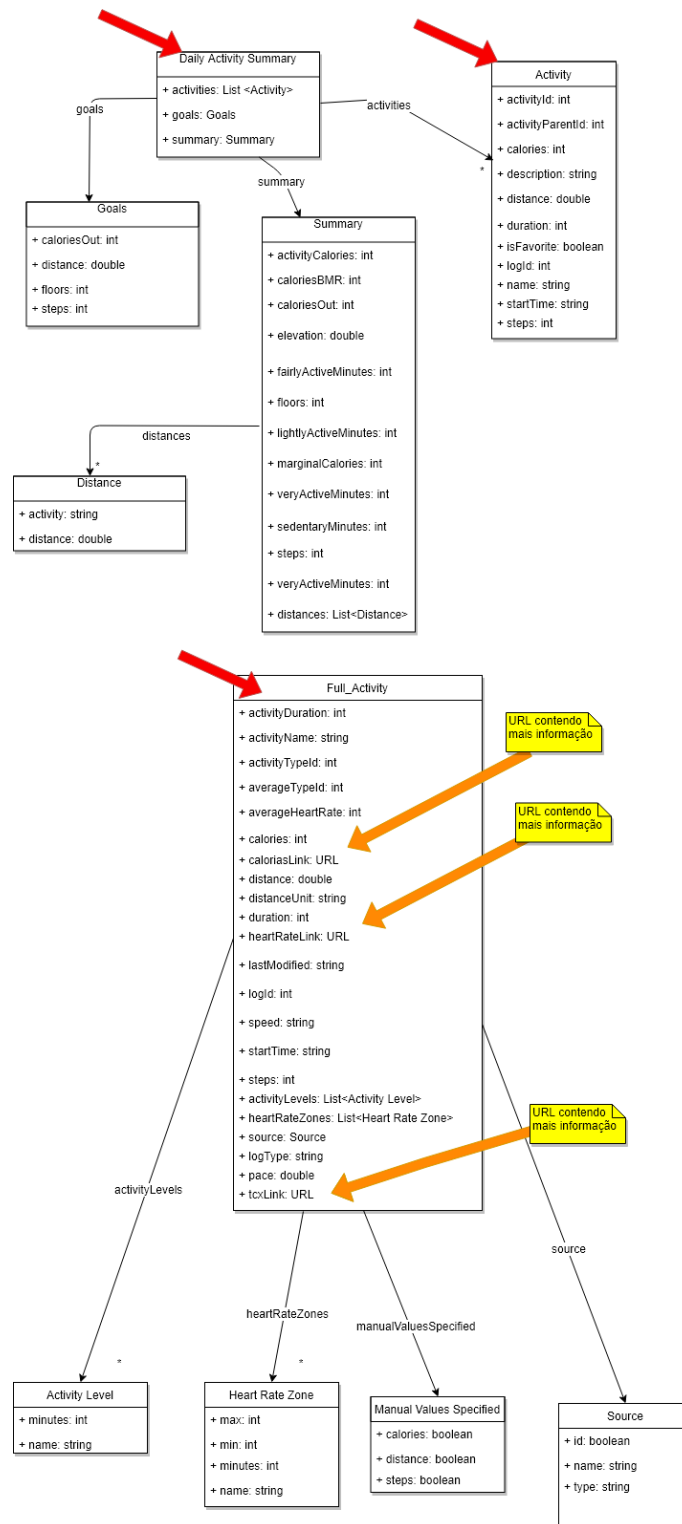


Figura A.2: Modelo de dados *Fitbit* - Atividade

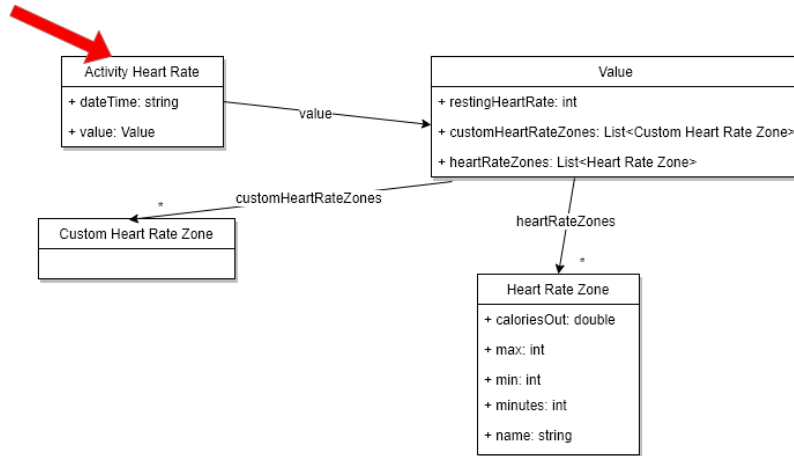


Figura A.3: Modelo de dados *Fitbit* - Batimento cardíaco

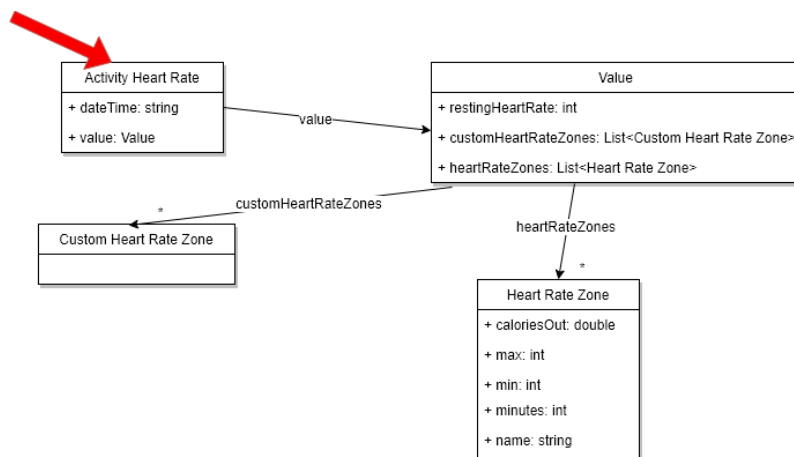


Figura A.4: Modelo de dados *Fitbit* - Corpo e Peso

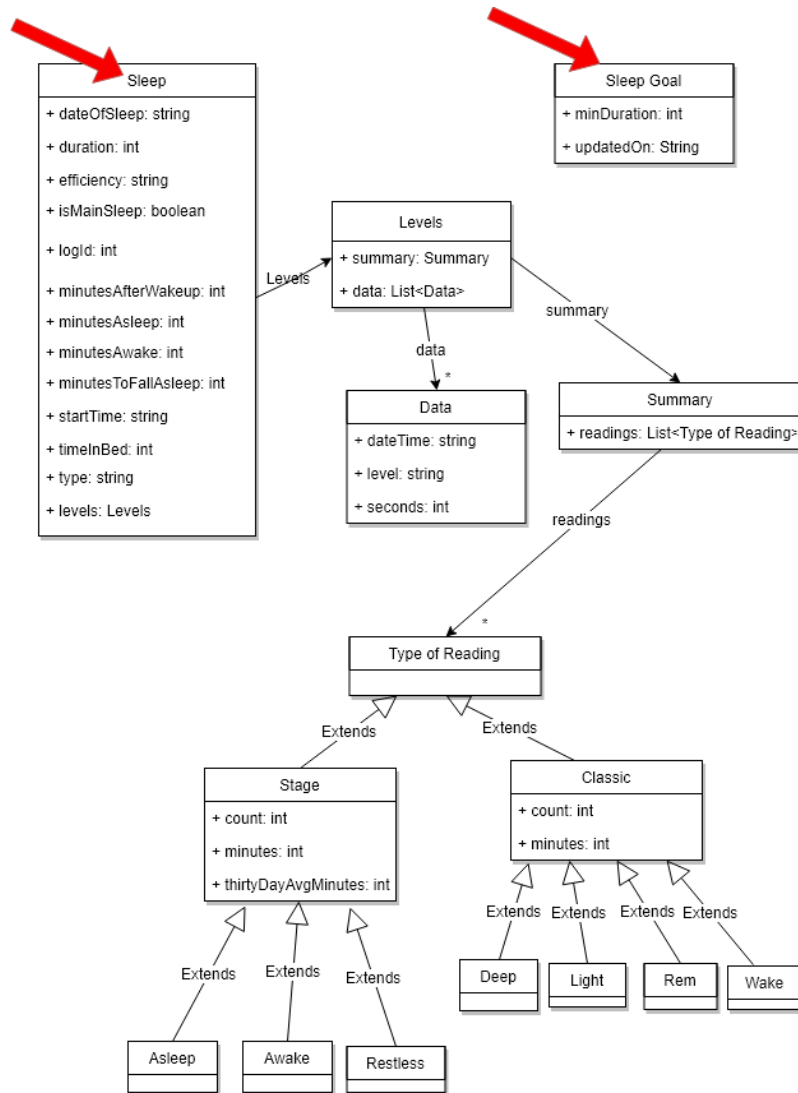


Figura A.5: Modelo de dados *Fitbit* - Sono

Anexo B

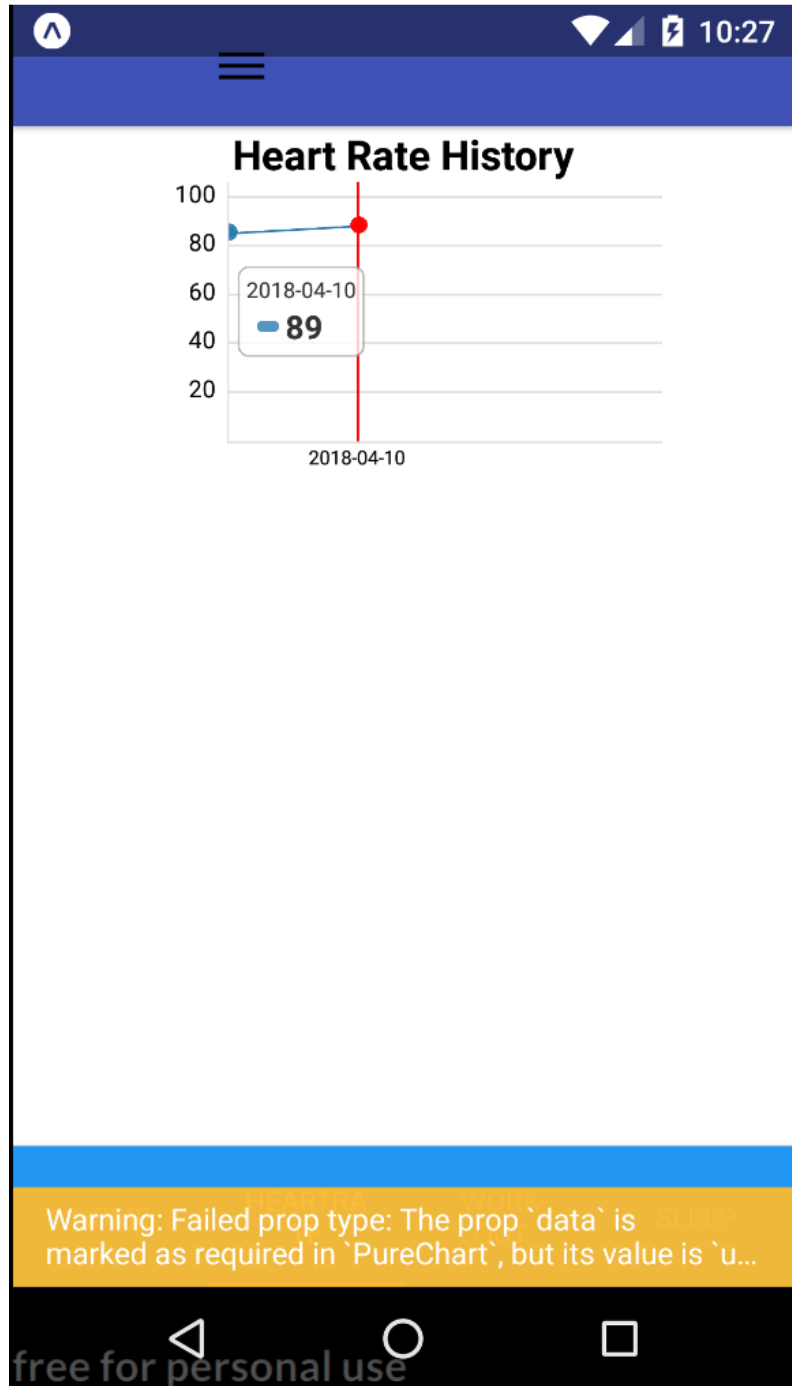
Resultado dos testes Unitários

```
P5 C:\Users\joao.m.gomes\Documents\Git\PGHD-REST-API\pghdbyg\intt-rest-api> node .\tests\test.js
TAP version 13
# User Data
Connected to mysql
ok 1 ID
ok 2 First Name
ok 3 LastName
# User Fitbit Config
ok 4 ID
ok 5 Fitbit ID
ok 6 Timezone
# User Google Fit Config
ok 7 ID
ok 8 Google Fit ID
ok 9 Locale
# Body Values - Fitbit
ok 10 ID
ok 11 Activity External ID
ok 12 BMI
ok 13 SOURCE
# Body Values - Google Fit
ok 14 ID
ok 15 Activity External ID
ok 16 BMI
ok 17 SOURCE
# Activity - Fitbit
ok 18 ID
ok 19 USER ID
ok 20 Activity External ID
ok 21 DESCRIPTION
ok 22 SOURCE
# Activity - Google Fit
ok 23 ID
ok 24 USER ID
ok 25 Activity External ID
ok 26 DESCRIPTION
ok 27 SOURCE
# Heart Rate
ok 28 ID
ok 29 USER ID
ok 30 SOURCE
# Heart Rate Zone
ok 31 ID
ok 32 MAX
ok 33 MIN
ok 34 MINUTES
# Sleep
ok 35 ID
ok 36 ID
ok 37 SOURCE
ok 38 DURATION
# Sleep Data
ok 39 ID
ok 40 DATE TIME
# Sleep Type of Reading
ok 41 ID
1..41
# tests 41
# pass 41
# ok
```

Figura B.1: Resultado dos testes unitários

Anexo C

Ecrãs da aplicação móvel

Figura C.1: Ecrã da funcionalidade *REQ-FUNCA*

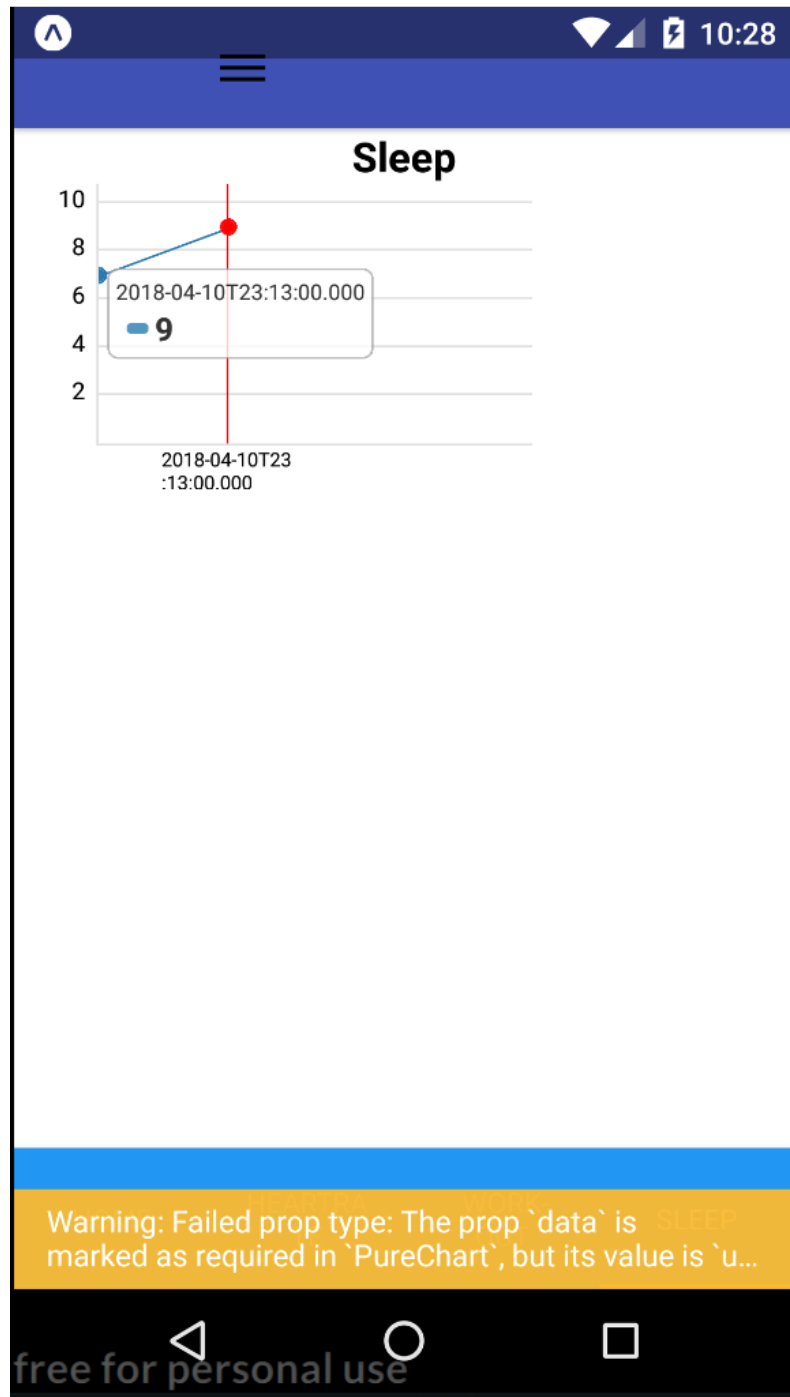


Figura C.2: Ecrã da funcionalidade *REQ-FUNC5*

Anexo D

Print-Screens da base de dados

UserConfigurationFitbit_UserID	googleFitbitExternalUserID	country	distanceUnit	footcandle	glucoseUnit	heightUnit	weightUnit	waterUnitName	timezone	minUnit	starDayOfWeek	locale	user_cri	dt_cri	user_updt	dt_updt
1	50@72	PT	METRIC	cm	METRIC	METRIC	METRIC	lit	Europe/Lisbon	METRIC	MONDAY	en_PT	NULL	NULL	NULL	NULL

Figura D.1: Tabela *Userconfigurationfitbit*

UserConfigurationGoogleFit_UserID	googleExternalUserID	locale	distanceUnit	weightUnit	user_cri	dt_cri	user_updt	dt_updt
1	109120381115932389059	pt-PT	meters	kg	NULL	NULL	NULL	NULL

Figura D.2: Tabela *Userconfigurationgooglefit*

DeviceID	Device_UserID	battery	externalID	deviceVersion	lastSyncTime	typeDevice	supplier	user_cri	dt_cri	user_updt	dt_updt
5	1	Emotv	568953324	MobileTrack	2018-04-13T16:25:31.000	TRACKER	FITBIT	NULL	NULL	NULL	NULL

Figura D.3: Tabela *Device*

HearRateValueId	HearRateValue_UserID	datetimeEntry	hearRateValue	sourceEntry	restingHeartRate	user_cri	dt_cri	user_updt	dt_updt
2	1	2018-04-09	86	FITBIT	0	NULL	NULL	NULL	NULL
3	1	2018-04-10	89	FITBIT	0	NULL	NULL	NULL	NULL

Figura D.4: Tabela *Heartratevalue*

HearRateZoneId	HearRateZone_HearRateValueID	maxEntry	minEntry	minutes	nameEntry	user_cri	dt_cri	user_updt	dt_updt
5	2	94	30	593	Out of ranoe	NULL	NULL	NULL	NULL
6	2	132	94	46	Fat Burn	NULL	NULL	NULL	NULL
7	2	160	132	0	Cardio	NULL	NULL	NULL	NULL
8	2	220	160	0	Peak	NULL	NULL	NULL	NULL
9	3	90	30	655	Out of ranoe	NULL	NULL	NULL	NULL
10	3	145	96	34	Fat Burn	NULL	NULL	NULL	NULL
11	3	172	134	0	Cardio	NULL	NULL	NULL	NULL
12	3	220	155	0	Peak	NULL	NULL	NULL	NULL

Figura D.5: Tabela *Heartratezone*

SleepId	Sleep_UserId	dateTime	durationInSeconds	efficiency	source	isHardSleep	externalSignal	minutesAfterWakeUp	minutesAsleep	minutesAwake	minutesTotalAsleep	startTime	endTime	levelEntry	typeEntry	user_cri	dt_cri	user_upd	dt_upd
1	1	2018-04-09	3200000	100	FBTBT	0000	179237270	0	420	0	0	2018-04-09T22:13:00.000	2018-04-10T05:13:00.000	420	asleep	0000	0000	0000	0000
2	1	2018-04-11	3240000	100	FBTBT	0000	179237291	0	540	0	0	2018-04-10T23:13:00.000	2018-04-10T23:13:00.000	540	asleep	0000	0000	0000	0000

Figura D.6: Tabela *Sleep*

SleepDataId	SleepData_SleepId	dateTimeEntry	levelEntry	seconds	user_cri	dt_cri	user_upd	dt_upd
1	1	2018-04-09T22:13:00.000	asleep	25200	NULL	NULL	NULL	NULL
2	2	2018-04-10T23:13:00.000	asleep	32400	NULL	NULL	NULL	NULL

Figura D.7: Tabela *SleepData*

SleepTypeOfReadingId	SleepTypeOfReading_SleepId	countEntry	minutes	thirtyDayAvgMinutes	typeOfReadingName	user_cri	dt_cri	user_upd	dt_upd
1	1	0	0	0	asleep	NULL	NULL	NULL	NULL
2	1	0	0	0	awake	NULL	NULL	NULL	NULL
3	1	0	0	0	restless	NULL	NULL	NULL	NULL
4	2	0	0	0	asleep	NULL	NULL	NULL	NULL
5	2	0	0	0	awake	NULL	NULL	NULL	NULL
6	2	0	0	0	restless	NULL	NULL	NULL	NULL

Figura D.8: Tabela *SleepTypeOfReading*

Anexo E

Fontes de dados *print-screens*

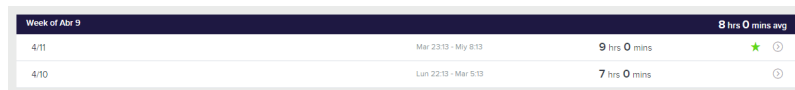


Figura E.1: Informação de sono da *Fitbit*



Figura E.2: Informação do utilizador da *Google Fit*

Referências

- [1] FEUP. Dissertação mesw. Disponível em https://sigarra.up.pt/feup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia_id=401041, acessado última vez no dia 18, Maio de 2018.
- [2] MESW. Mesw. Disponível em https://sigarra.up.pt/feup/pt/cur_geral.cur_view?pv_curso_id=10861&pv_origem=CAND, acessado última vez no dia 18, Maio de 2018.
- [3] FEUP. Feup. Disponível em https://sigarra.up.pt/feup/pt/web_page.inicial, acessado última vez no dia 18, Maio de 2018.
- [4] Glintt. Glintt homepage. glintt.com, Acessado pela última vez no dia 18, Maio de 2018.
- [5] Oxford Dictionary. Wearable definition. Disponível em <https://en.oxforddictionaries.com/definition/wearable>, Acessado pela última vez no dia 5, Dezembro 2017.
- [6] Best Buy. Fitbit charge 2. Disponível em <https://multimedia.bbcastatic.ca/multimedia/products/500x500/104/10481/10481848.jpg>, Acessado pela última vez no dia 5, Dezembro 2017.
- [7] Ariel Adams. Apple smartwatch. <https://www.ablogtowatch.com/wp-content/uploads/2014/09/Apple-Watch-37.jpg>, Acessado pela última vez no dia 5, Dezembro 2017.
- [8] Tandem Diabetes. Dexcom g5. Disponível em https://www.tandemdiabetes.com/images/default-source/2300-x-1440/2300x1440_dexcom_difference.jpg?sfvrsn=28ee3cd7_2&size=1280, Acessado pela última vez no dia 5, Dezembro 2017.
- [9] Sarah Griffiths. Alex posture. Disponível em http://i.dailymail.co.uk/i/pix/2016/01/25/13/308EA1E000000578-0-image-a-62_1453726814556.jpg, Acessado pela última vez no dia 5, December 2017.
- [10] Google. Google trends on wearables. Disponível em <https://trends.google.pt/trends/explore?date=today%205-y&q=wearables>, Acessado pela última vez no dia 20, Novembro 2017.
- [11] Statista. Number of connected wearable devices worldwide from 2016 to 2021 (in millions). Disponível em <https://www.statista.com/statistics/487291/global-connected-wearable-devices/>, Acessado pela última vez no dia 20, Novembro 2017.

- [12] Ogilvy and Mather. Wearable market segments. Disponível em http://www.emberify.com/blog/wp-content/uploads/2014/12/emberify_wearable_1.jpg, Acessado pela última vez no dia 20, Novembro 2017.
- [13] Statista. Wearable device market value. Disponível em http://www.emberify.com/blog/wp-content/uploads/2014/12/emberify_wearable_3.png, Acessado pela última vez no dia 20, Novembro 2017.
- [14] Fitbit. Fitbit official site for activity trackers and more. Disponível em <https://www.fitbit.com/uk/home>, acessado última vez no dia 19, Março de 2018.
- [15] Garmin. Garmin | portugal | home. Disponível em <http://www.garmin.com/pt-PT>, acessado última vez no dia 19, Março de 2018.
- [16] Google-Wear. Wear os by google smartwatches. Disponível em <https://wearos.google.com/>, acessado última vez no dia 19, Março de 2018.
- [17] Apple. ios - health - apple. Disponível em <https://www.apple.com/ios/health/>, acessado última vez no dia 19, Março de 2018.
- [18] Samsung. Samsung web page. Disponível em <http://www.samsung.com/pt/>, acessado última vez no dia 23, Março de 2018.
- [19] Kinesis. Kinesis qtug. Disponível em <https://www.kinesis.ie/qtug/>, Acessado pela última vez no dia 5, Dezembro 2017.
- [20] Fitbit. Fitbit charge 2 heart rate + fitness wristband. Disponível em <https://www.fitbit.com/uk/charge2>, acessado última vez no dia 19, Março de 2018.
- [21] Brian Otis and Babak Parviz. Official google blog: Introducing our smart contact lens project. Disponível em <http://googleblog.blogspot.pt/2014/01/introducing-our-smart-contact-lens.html>, acessado última vez no dia 19, Março de 2018.
- [22] Valedo. Valedo web page. Disponível em https://www.valedotherapy.com/de_en/, acessado última vez no dia 26, Março de 2018.
- [23] FDA. Medical device definition. Disponível em <https://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/ClassifyYourDevice/ucm051512.htm>, Acessado pela última vez no dia 20, Novembro 2017.
- [24] Christian Zibreg. Fda clarifies when health and fitness wearables should be considered medical devices.
- [25] FDA. Premarket notification 510(k). Disponível em <https://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/HowtoMarketYourDevice/PremarketSubmissions/PremarketNotification510k/default.htm>, Acessado pela última vez no dia 20, Novembro 2017.
- [26] FDA. De novo process. Disponível em <https://www.fda.gov/AboutFDA/CentersOffices/OfficeofMedicalProductsandTobacco/CDRH/CDRHTransparency/ucm232269.htm>, Acessado pela última vez no dia 20, Novembro 2017.

- [27] FDA. Designating humanitarian use device (hud). Disponível em <https://www.fda.gov/ForIndustry/DevelopingProductsforRareDiseasesConditions/DesignatingHumanitarianUseDevicesHUDS/default.htm>, Acedido pela última vez no dia 20, Novembro 2017.
- [28] FDA. Pre-market approval (pma). Disponível em <https://www.fda.gov/medicaldevices/deviceregulationandguidance/howtomarketyourdevice/premarket/submissions/premarketapprovalpma/>, Acedido pela última vez no dia 20, Novembro 2017.
- [29] FDA. Reclassification fda. Disponível em <https://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/Overview/ClassifyYourDevice/ucm080412.htm>, Acedido pela última vez no dia 20, Novembro 2017.
- [30] Michael J Hoffmann. Fda regulatory process michael.
- [31] European Parliament and of the Council. Council Directive 93/42/EEC. *Official Journal of the European Union*, (June 1993):1–60, 2007.
- [32] M.E. Whitman and H.J. Mattord. *Principles of Information Security*. Cengage Learning, 2011.
- [33] Preston Hogue. How to ensure the availability, integrity, and confidentiality of your apps.
- [34] M.S. Merkow and J. Breithaupt. *Information Security: Principles and Practices*. Always Learning. Pearson, 2014.
- [35] ISO. ISO/IEC 27002:2013. Technical report, 2013.
- [36] ISO. ISO 27799:2016 . Technical report, 2016.
- [37] Allen & Overy. The EU General Data Protection Regulation. page 12, 2017.
- [38] Deloitte. General Data Protection Regulation (GDPR). 2017.
- [39] European Union. Regulation 2016/679 of the European parliament and the Council of the European Union. *Official Journal of the European Communities*, 2014(October 1995):1–88, 2016.
- [40] Directive 2016/680 of the European Parliament and the Council on the protection of natural persons with regard to the processing of personal data by competent authorities for the purposes of the prevention, investigation, detection or prosecution of crimi. *Official Journal of the European Union*, 2014(April):L 119/89–L 119/131, 2016.
- [41] Article 29 Data Protection Working Party. Guidelines on the right to data portability. *October*, (April):1–11, 2003.
- [42] Directive 2013/37/EU of the European Parliament and of the Council of 26 June 2013 amending Directive 2003/98/EC on the re-use of public sector information. *Official Journal of the European Union*, pages 1–8, 2013.
- [43] ONC. Patient-generated health data. Disponível em <https://www.healthit.gov/policy-researchers-implementers/patient-generated-health-data>, Acedido última vez no dia 20, Fevereiro de 2018.

- [44] Patient-Generated Health Data. pages 1–35, 2012.
- [45] Clear Data. Patient generated health data: Challenges and opportunities. Disponível em <https://www.cleardata.com/blog/patient-generated-health-data-challenges-opportunities>, Acessado última vez no dia 20, Fevereiro de 2018.
- [46] ONC. Patient-generated health data. Disponível em <http://www.aappublications.org/news/2017/03/06/HIT030617>, Acessado última vez no dia 20, Fevereiro de 2018.
- [47] Formfast. The role eforms play in patient-generated healthcare data (pghd). Disponível em <https://formfast.com/pghd-formfast-connect/>, acessado última vez no dia 21, Fevereiro de 2018.
- [48] Apple. Apple web page. Disponível em <https://www.apple.com/pt/>, acessado última vez no dia 26, Março de 2018.
- [49] Apple. Apple healthkit. Disponível em <https://developer.apple.com/healthkit/>, acessado última vez no dia 26, Fevereiro de 2018.
- [50] Epic. Epic haiku. Disponível em <https://itunes.apple.com/us/app/epic-haiku/id348308661?mt=8>, acessado última vez no dia 26, Fevereiro de 2018.
- [51] Apple. Apple researchkit. Disponível em <https://www.apple.com/researchkit/>, acessado última vez no dia 26, Fevereiro de 2018.
- [52] Adnan Farooqui. S-patch3 wearable with samsung's bio processor spotted at fcc. Disponível em <https://www.sammobile.com/2017/06/07/s-patch3-wearable-with-samsungs-bio-processor-spotted-at-fcc/>, acessado última vez no dia 19, Março de 2018.
- [53] Up Right. Up right web page. Disponível em https://www.uprightpose.com/?utm_source=uprightgo.com&utm_medium=Referral, acessado última vez no dia 26, Março de 2018.
- [54] Alex Posture. Alex posture web page. Disponível em <https://alexposture.com/>, acessado última vez no dia 26, Março de 2018.
- [55] Zitko. Zitko walk web page. Disponível em <http://www.zikto.com/main.html>, acessado última vez no dia 26, Março de 2018.
- [56] Lumo Body Tech. Lumo lift web page. Disponível em <https://www.lumobodytech.com/lumo-lift/>, acessado última vez no dia 26, Março de 2018.
- [57] Lumo Body Tech. Lumo developers web page. Disponível em <https://dev.lumobodytech.com/>, acessado última vez no dia 26, Março de 2018.
- [58] PK Vitality. K'watch glucose web page. Disponível em <http://www.pkvitality.com/ktrack-glucose/>, acessado última vez no dia 26, Março de 2018.
- [59] FreeStyle Diabetes. Freestyle libre sensor web page. Disponível em <https://freestylediabetes.co.uk/freestyle-libre/freestyle-libre-sensor>, acessado última vez no dia 26, Março de 2018.

- [60] Dexcom. Dexcom developers web page. Disponível em <https://developer.dexcom.com/>, acessado última vez no dia 26, Março de 2018.
- [61] Statista. Wearables market share companies 2014-2017.
- [62] Google. Google fit web page. Disponível em <https://www.google.com/fit/>, acessado última vez no dia 20, Março de 2018.
- [63] Statista. Most popular health and fitness apps in the united states as of july 2017, by monthly active users (in millions).
- [64] Fitbit. Fitbit google store. Disponível em https://play.google.com/store/apps/details?id=com.fitbit.FitbitMobile&hl=pt_PT, acessado última vez no dia 26, Março de 2018.
- [65] Google. Google fit google store. Disponível em https://play.google.com/store/apps/details?id=com.google.android.apps.fitness&hl=pt_PT, acessado última vez no dia 26, Março de 2018.
- [66] Samsung. Samsung health google store. Disponível em https://play.google.com/store/apps/details?id=com.sec.android.app.shealth&hl=pt_PT, acessado última vez no dia 26, Março de 2018.
- [67] Fitbit. Help article: What should i know about sleep stages? Disponível em https://help.fitbit.com/articles/en_US/Help_article/2163/?l=en_US&c=Topics%3ASleep&fs=Search&pn=1, acessado última vez no dia 26, Março de 2018.
- [68] Garmin. Garmin api. <https://developer.garmin.com/garmin-connect-api/overview/>, Acessado pela última vez no dia 18, Maio de 2018.
- [69] Tizen. Tizen web page. Disponível em <https://www.tizen.org/>, acessado última vez no dia 20, Março de 2018.
- [70] Lose It. Lose it web page. Disponível em <https://www.loseit.com/>, acessado última vez no dia 21, Março de 2018.
- [71] Armour Record. Under armour record web page. Disponível em <https://record.underarmour.com/>, acessado última vez no dia 21, Março de 2018.
- [72] Run Keeper. Run keeper web page. Disponível em <https://runkeeper.com/>, acessado última vez no dia 21, Março de 2018.
- [73] Nike Run Club. Nike run club web page. Disponível em https://www.nike.com/pt/pt_pt/c/running/nike-run-club, acessado última vez no dia 21, Março de 2018.
- [74] Under Armour. My fitnesspal web page. Disponível em <https://www.myfitnesspal.com/>, acessado última vez no dia 21, Março de 2018.
- [75] Azumio. Instant heart rate by azumio web page. Disponível em <http://www.azumio.com/s/instantheartrate/index.html>, acessado última vez no dia 21, Março de 2018.
- [76] Nokia. Nokia health mate web page. Disponível em <https://health.nokia.com/pt/en/health-mate>, acessado última vez no dia 21, Março de 2018.

- [77] Nokia. Smartband 2 swr12 web page. Disponível em <https://play.google.com/store/apps/details?id=com.sonymobile.hostapp.everest>, acessido última vez no dia 21, Março de 2018.
- [78] Google. Google wear web page. Disponível em <https://wearos.google.com/>, acessido última vez no dia 21, Março de 2018.
- [79] Fossil. Fossil q control web page. Disponível em <https://www.fossil.com/us/en/wearable-technology/smartwatches/q-control-learn-more.html>, acessido última vez no dia 21, Março de 2018.
- [80] Huawei. Huawei watch 2 web page. Disponível em <https://consumer.huawei.com/pt/wearables/watch2/>, acessido última vez no dia 21, Março de 2018.
- [81] New Balance. New balance runiq web page. Disponível em <https://www.newbalance.co.nz/featured/collections/runiq/>, acessido última vez no dia 21, Março de 2018.
- [82] LG. Lg watch sport web page. Disponível em <http://www.lg.com/us/smart-watches/lg-W280A-sport>, acessido última vez no dia 21, Março de 2018.
- [83] Polar. Polar m600 web page. Disponível em <https://www.polar.com/pt/produtos/sport/M600-smartwatch-com-GPS>, acessido última vez no dia 21, Março de 2018.
- [84] Motorola. Moto 360 sport web page. Disponível em <https://www.motorola.com.au/products/moto-360-sport>, acessido última vez no dia 21, Março de 2018.
- [85] Google. Google android api web page. Disponível em <https://developers.google.com/fit/android/>, acessido última vez no dia 21, Março de 2018.
- [86] Google. Google rest api web page. Disponível em <https://developers.google.com/fit/rest/>, acessido última vez no dia 21, Março de 2018.
- [87] Fitbit. Fitbit sdk web page. Disponível em <https://dev.fitbit.com/>, acessido última vez no dia 21, Março de 2018.
- [88] Fitbit. Fitbit studio web page. Disponível em <https://studio.fitbit.com/>, acessido última vez no dia 21, Março de 2018.
- [89] Fitbit. Fitbit apis reference web page. Disponível em <https://dev.fitbit.com/build/reference/web-api/>, acessido última vez no dia 21, Março de 2018.
- [90] Apple. Watch os web page. Disponível em <https://www.apple.com/pt/watchos/>, acessido última vez no dia 21, Março de 2018.
- [91] LinuxFoundation. Linuxfoundation web page. Disponível em <http://www.linuxfoundation.org/>, acessido última vez no dia 20, Março de 2018.
- [92] Tizen. Tizen application developers web page. Disponível em <https://developer.tizen.org/>, acessido última vez no dia 20, Março de 2018.
- [93] Tizen. Tizen platform developers web page. Disponível em <https://source.tizen.org/>, acessido última vez no dia 20, Março de 2018.

- [94] Samsung. Samsung health server sdk. Disponível em <http://developer.samsung.com/health/server>, acessido última vez no dia 23, Março de 2018.
- [95] Samsung. Samsung health. Disponível em <http://www.samsung.com/pt/apps/samsung-health/>, acessido última vez no dia 23, Março de 2018.
- [96] Samsung. Apply for partner | samsung health server sdk. Disponível em <https://developer.samsung.com/health/server/apply-for-partner>, acessido última vez no dia 02, Maio de 2018.
- [97] Wearable. Best fitbit 2018: All our fitbit reviews for you to compare. Disponível em <https://www.wearable.com/fitbit/what-fitbit-tracker-should-you-buy>, acessido última vez no dia 04, Maio de 2018.
- [98] Fitbit. Fitbit tracker comparison. Disponível em <https://www.fitbit.com/compare>, acessido última vez no dia 27, Fevereiro de 2018.
- [99] J. Webber, S. Parastatidis, and I. Robinson. *REST in Practice: Hypermedia and Systems Architecture*. Theory in practice series. O'Reilly Media, 2010.
- [100] D. Gourley, B. Totty, M. Sayer, A. Aggarwal, and S. Reddy. *HTTP: The Definitive Guide: The Definitive Guide*. Definitive Guides. O'Reilly Media, 2002.
- [101] J. Richer and A. Sanso. *OAuth 2 in Action*. Manning Publications, 2017.
- [102] Google. Api reference google fit. Disponível em <https://developers.google.com/fit/rest/v1/reference/>, acessido última vez no dia 02, Maio de 2018.
- [103] Google. Sessions | google fit | google developers. Disponível em <https://developers.google.com/fit/rest/v1/sessions>, acessido última vez no dia 02, Maio de 2018.
- [104] Fitbit. Activity and exercise logs. Disponível em <https://dev.fitbit.com/build/reference/web-api/activity/>, acessido última vez no dia 02, Maio de 2018.
- [105] Samsung. Data access | samsung developers user endpoint. Disponível em <https://developer.samsung.com/health/server/api/data-access#user-profile-api>, acessido última vez no dia 08, Maio de 2018.
- [106] Samsung. Data access | samsung developers devices endpoint. Disponível em <https://developer.samsung.com/health/server/api/data-access#device-api>, acessido última vez no dia 08, Maio de 2018.
- [107] Samsung. Data types_weight | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/weight>, acessido última vez no dia 08, Maio de 2018.
- [108] Samsung. Data types_heart_rate | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/heart-rate>, acessido última vez no dia 08, Maio de 2018.
- [109] Samsung. Data types_exercise | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/exercise>, acessido última vez no dia 08, Maio de 2018.

- [110] Samsung. Data types_sleep | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/sleep>, acessido última vez no dia 08, Maio de 2018.
- [111] Samsung. Data types_sleep_stage | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/sleep-stage>, acessido última vez no dia 08, Maio de 2018.
- [112] Samsung. Data types_body_temperature | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/body-temperature>, acessido última vez no dia 08, Maio de 2018.
- [113] Samsung. Data types_oxygen_saturation | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/oxygen-saturation>, acessido última vez no dia 08, Maio de 2018.
- [114] Samsung. Data types_electrocardiogram | samsung developers. Disponível em <https://developer.samsung.com/health/server/api/data-types/electrocardiogram>, acessido última vez no dia 08, Maio de 2018.
- [115] CUF. Ecg - eletrocardiograma. Disponível em <https://www.saudecuf.pt/areas-clinicas/exames/cardiologia/ecg-eletrocardiograma>, acessido última vez no dia 08, Maio de 2018.
- [116] B. Karwin. *SQL Antipatterns: Avoiding the Pitfalls of Database Programming*. Pragmatic Bookshelf Series. Pragmatic Bookshelf, 2010.
- [117] Xplenty. The sql vs nosql difference: Mysql vs mongodb. Disponível em <https://medium.com/xplenty-blog/the-sql-vs-nosql-difference-mysql-vs-mongodb-32c9980e67b2>, acessido última vez no dia 08, Maio de 2018.
- [118] Mike Cohn. User stories and user story examples by mike cohn. Disponível em <https://www.mountaingoatsoftware.com/agile/user-stories>, acessido última vez no dia 16, Abril de 2018.
- [119] Facebook. React native page. Disponível em <https://facebook.github.io/react-native/>, acessido última vez no dia 11, Maio de 2018.
- [120] Ryan Dahl. Node.js page. Disponível em <https://nodejs.org/en/>, acessido última vez no dia 11, Maio de 2018.
- [121] Oracle Corporation. Mysql page. Disponível em <https://www.mysql.com/>, acessido última vez no dia 11, Maio de 2018.
- [122] Javascript. Javascript page. Disponível em <https://www.javascript.com/>, acessido última vez no dia 11, Maio de 2018.
- [123] Facebook. React page. Disponível em <https://reactjs.org/>, acessido última vez no dia 11, Maio de 2018.
- [124] Facebook. Facebook developers. Disponível em <https://developers.facebook.com/>, acessido última vez no dia 11, Maio de 2018.

- [125] Renato Probst. Vantagens da utilização do react native no desenvolvimento de aplicativos android e ios. Disponível em <http://www.rockapps.com.br/vantagens-react-native-desenvolvimento-de-aplicativos-android-e-ios/>, acessado última vez no dia 11, Maio de 2018.
- [126] Facebook. Expo page. Disponível em <https://expo.io/>, acessado última vez no dia 11, Maio de 2018.
- [127] Joshua McNeese. Introduction to node.js.
- [128] Node.js. Npm page. Disponível em <https://www.npmjs.com/>, acessado última vez no dia 11, Maio de 2018.
- [129] Oracle Corporation. Mysql documentation. Disponível em <https://dev.mysql.com/doc/>, acessado última vez no dia 11, Maio de 2018.
- [130] John Terzakis. Specifying Effective Non-Functional Requirements. pages 1–64, 2012.
- [131] R.B. Grady. *Practical Software Metrics for Project Management and Process Improvement*. Prentice Hall, 1992.
- [132] MySQL. Mysql workbench page. Disponível em <https://www.mysql.com/products/workbench/>, acessado última vez no dia 11, Maio de 2018.