



INSTITUTO POLITÉCNICO DE COIMBRA
INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

**Sistema de monitorização para serviço de urgência em
ambiente hospitalar - Estágio na Sensing Future**

**Relatório de Projeto para a obtenção do grau de Mestre em
Instrumentação Biomédica**

Autor

Adriana Patrícia Monteiro Reis

Orientação

Prof.^a Dr.^a Fernanda de Madureira Coutinho
Prof. Dr. João Paulo Morais Ferreira

Sistema de monitorização para serviço de urgência em ambiente hospitalar

Relatório de Estágio apresentado para a obtenção do grau de Mestre em Instrumentação Biomédica

Autora

Adriana Patrícia Monteiro Reis

Orientadores

Prof.^a Dr.^a Fernanda de Madureira Coutinho

Prof. Dr. João Paulo Morais Ferreira

Professores do Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia de Coimbra

Supervisores

Me. Luís André Gonçalves Ferreira

Sensing Future Technologies

Me. João Manuel Leitão Quintas

Laboratório de Automática e Sistemas do Instituto Pedro Nunes

Coimbra, julho de 2017

Resumo

O serviço de urgência dos hospitais continua a ser a primeira opção de grande maioria dos doentes. Na realidade, a maioria das causas que levam a que os doentes recorram a este tipo de serviços, de acordo com o protocolo de triagem de Manchester, não é urgente ou é pouco urgente, pelo que o doente poderia ser tratado em unidades de cuidados primários, evitando a ida ao hospital. A grande afluência aos serviços de urgência dos hospitais, aliada à insuficiência de profissionais de saúde nestes serviços, faz com que, muitas vezes, o doente se encontre sujeito a tempos de espera demasiado longos. Por consequência, o seu estado de saúde pode agravar-se após a triagem e até que este seja observado pelo médico da especialidade, o que faz com que a sua condição de saúde possa ser colocada em risco sem que o profissional de saúde se possa aperceber e agir atempadamente.

Atualmente, existem diversas soluções de *hardware* capazes de monitorizar alguns parâmetros vitais, bem como soluções de localização *indoor* com recurso a diferentes métodos e tecnologias. No entanto, não existe nenhuma solução que englobe estas duas vertentes e que seja aplicável a um serviço de urgência hospitalar. Este enquadramento constituiu a motivação principal para o trabalho desenvolvido neste estágio, particularmente a criação de um sistema capaz de monitorizar os parâmetros vitais do doente que recorre às urgências, nomeadamente a frequência cardíaca e a temperatura corporal, bem como a sua localização neste serviço, e capaz de dar alertas, de forma a auxiliar o profissional de saúde a detetar situações de agravamento da condição de saúde do doente.

No protótipo desenvolvido, é colocada uma pulseira no doente, que faz a leitura dos seus sinais vitais (frequência cardíaca e temperatura corporal), devendo este ser acompanhado de um telemóvel com uma aplicação *Android* específica, desenvolvida para enviar os dados dos sinais vitais recolhidos, assim como a informação da localização do doente, periodicamente, para um servidor. Neste, há uma base de dados, com a informação de todos os doentes sob atendimento e, através de uma *dashboard*, é feito o interface com o profissional de saúde, alertando-o, em caso de necessidade, sempre que ocorra alguma situação considerada crítica. O protótipo foi avaliado na perspetiva dos profissionais de saúde, realizada com recurso a testes de usabilidade, e também numa perspetiva empresarial, levada a cabo junto das empresas *Sensing Future Technologies* e *Glintt*, e pelo Laboratório de Automática e Sistemas do Instituto Pedro Nunes.

Palavras-chave: serviços de urgência, triagem de Manchester, sinais vitais, localização *indoor*, monitorização, instrumentação biomédica.

Abstract

The hospital emergency service remains the first choice of the vast majority of patients. In fact, most of the causes that lead patients to use this type of services, according to the Manchester triage protocol, is not urgent or is not very urgent, so that the patient could be treated in primary care units, avoiding going to the hospital. The large influx of hospital emergency services, combined with the lack of health professionals, often makes the patients susceptible to excessive waiting times. As a result, their health condition may worsen after triage and until they are observed by the doctor of the specialty, which causes their health condition to be put at risk without the health professional being able to perceive and act in a timely manner.

Currently there are several hardware solutions that can monitor some vital parameters as well as indoor location solutions using different methods and technologies. However, there is no solution that encompasses these two concerns and is applicable to hospital emergency department. This issue was the main motivation for the work carried out at this internship, particularly the creation of a system capable of monitoring the vital parameters of the patients, such as heart rate and body temperature, as well as their location in this service, and to give alerts, in order to help the health professional to detect situations of worsening of the patient's health condition.

In the developed prototype, a bracelet is placed on the patient's wrist, which reads their vital signs (heart rate and body temperature), accompanied by a mobile phone with an Android application, designed to send the collected vital signs data, as well as patient's location, periodically, to a server. In this, there is a database, with the information of all the patients under care and, through a dashboard, the interface is made with the health professional, alerting him, if necessary, whenever a situation is considered critical. The prototype was evaluated from the perspective of health professionals, performed using usability tests, and also from a business perspective, carried out by the companies Sensing Future Technologies and Glintt, and by the Laboratório de Automática e Sistemas of Instituto Pedro Nunes.

Keywords: emergency services, Manchester triage, vital signs, indoor location, monitoring, biomedical instrumentation.

Agradecimentos

Em primeiro lugar, deixo expresso o meu agradecimento aos meus pais, por todas as experiências que me proporcionaram a nível pessoal e pela oportunidade que me deram de frequentar, não só a Licenciatura em Engenharia Biomédica, como a Licenciatura em Engenharia Eletrotécnica e o Mestrado em Instrumentação Biomédica. Obrigada por estarem ao meu lado nesta caminhada e me incentivarem a fazer sempre mais e melhor. Aos meus avós paternos e maternos, à minha avó, que me ajudou a educar e a crescer, e ao meu avô, que não conseguiu terminar comigo este percurso, muito obrigada. Agradeço ainda ao meu irmão, para quem tento sempre ser um bom exemplo, e ao meu namorado e melhor amigo, que celebrou comigo todas as vitórias e me ajudou a superar as derrotas.

Agradeço à Prof.^a Dr.^a Fernanda Coutinho e ao Prof. Dr. João Ferreira, que desde logo se mostraram disponíveis para orientar este estágio, e que me ajudaram bastante na elaboração deste relatório. Foram uma presença constante durante o estágio, e sem dúvida me proporcionaram uma excelente orientação.

Quero também agradecer à *Sensing Future Technologies* e ao Laboratório de Automática e Sistemas por me ter proporcionado este estágio, por tudo o que aprendi, por tudo o que tive oportunidade de explorar, mas acima de tudo pela experiência em si, pelo que cresci a nível pessoal e profissional. Agradeço em especial ao Me. Luís Ferreira e ao Me. João Quintas, que se mostraram sempre disponíveis para me ajudar a ultrapassar os obstáculos que foram surgindo, e ao Me. André Dias, com quem aprendi muito.

Por fim, deixo o meu agradecimento a todos os professores e colegas de curso do Instituto Superior de Engenharia de Coimbra com quem me cruzei no decurso do Mestrado, e que contribuíram para a minha formação profissional e pessoal, e a todos os meus amigos que de alguma forma participaram da minha formação académica.

Índice

Resumo	i
<i>Abstract</i>	iii
Agradecimentos	v
Lista de figuras	ix
Lista de tabelas	xiii
Lista de acrónimos.....	xv
1. Introdução.....	1
1.1 Enquadramento	1
1.2 Motivação	1
1.3 Objetivos.....	2
1.4 Empresa de acolhimento	3
1.5 Organização do relatório.....	3
2. Estado da arte	5
2.1 Serviços de urgência	5
2.1.1 Funcionamento típico de um serviço de urgência	6
2.1.2 Triagem de Manchester	6
2.1.3 Serviços de urgência em Portugal	8
2.1.4 Mortalidade nos serviços de urgência portugueses	9
2.2 Sistemas de monitorização de sinais vitais	10
2.2.1 Métodos de medição de sinais vitais	10
2.2.2 Importância dos sinais vitais na triagem.....	16
2.2.3 Sistemas comerciais e patentes de dispositivos de medição de sinais vitais	16
2.3 Localização <i>indoor</i>	18
2.3.1 Métodos de localização <i>indoor</i>	19
2.3.2 Tecnologias usadas em sistemas de localização <i>indoor</i>	21
2.3.3 Comparação entre métodos e tecnologias usados em sistemas de localização <i>indoor</i>	25
2.3.4 Sistemas comerciais e patentes de sistemas de localização <i>indoor</i>	27
2.4 Considerações finais	28
3. Requisitos do sistema e arquitetura	31
3.1 Identificação dos requisitos	31
3.2 <i>User stories</i>	32
3.2.1 Funcionalidades sob a perspetiva do doente.....	33
3.2.2 Funcionalidades sob a perspetiva do profissional de saúde	33
3.3 Especificações técnicas do <i>software</i>	34
3.3.1 Interface inicial	35
3.3.2 Inserir novo doente	35

3.3.3	Lista de doentes.....	36
3.3.4	Informação do doente.....	37
3.4	Linguagens de programação.....	37
3.5	Esquemas de arquitetura.....	38
3.5.1	Diagrama de componentes	38
3.5.2	Diagrama de atividades	39
3.5.3	Diagrama de sequência	41
3.6	Especificações técnicas do <i>hardware</i>	44
3.7	Considerações finais.....	45
4.	Descrição da solução implementada.....	47
4.1	Etapas do processo de desenvolvimento	47
4.2	Seleção de <i>software</i>	48
4.2.1	Linguagem de programação PHP	49
4.2.2	<i>Frameworks</i> e bibliotecas	49
4.3	Seleção de <i>hardware</i>	52
4.3.1	Avaliação e seleção de <i>hardware</i>	53
4.4	Aplicação <i>Android</i> de recolha de dados	55
4.4.1	Desenvolvimento de uma aplicação <i>Android</i> para a <i>Microsoft Band 2</i>	56
4.4.2	Integração do algoritmo de localização <i>indoor</i> na aplicação <i>Android</i> desenvolvida...	59
4.5	<i>Dashboard</i>	65
4.5.1	Conceção da base de dados	66
4.5.2	Implementação da <i>dashboard</i>	72
4.5.3	Ações responsivas	82
4.6	Comunicação entre a aplicação <i>Android</i> e a base de dados	83
4.6.1	REST API	83
4.6.2	SOAP	84
4.6.3	Diferenças entre REST e SOAP.....	85
4.7	Considerações finais.....	90
5.	Avaliação do protótipo desenvolvido	91
5.1	Descrição.....	91
5.2	Modo de utilização	92
5.3	Avaliação do protótipo na perspetiva do profissional de saúde	94
5.3.1	Testes de usabilidade	94
5.3.2	Discussão dos resultados.....	96
5.4	Avaliação do protótipo na perspetiva empresarial	98
6.	Conclusões e trabalhos futuros	103
	Bibliografia	105

Lista de figuras

Figura 1 - Atribuição de cores das pulseiras na Triagem de Manchester (Fonte:(Gomes, 2013)).	7
Figura 2 - Tempos de espera de acordo com a Triagem de Manchester (Fonte: (“Hospitais a rebentar pelas costuras”, 2014)).	8
Figura 3 - Tipos de termómetros. (a) Termómetro usado para medir a temperatura do tímpano. (b) Termómetro para medir a temperatura superficial da pele (Fonte: <i>Google</i> Imagens).	11
Figura 4 - Oxímetro de pulso (Fonte: <i>Google</i> Imagens).	15
Figura 5 – Técnicas de trilateração. (a) Trilateração 2-D. (b) Trilateração 3-D (Fonte: (Circuit Cellar Staff, 2014)).	20
Figura 6 – <i>Mockup</i> da interface inicial da <i>dashboard</i> .	35
Figura 7 – <i>Mockup</i> da interface que permite inserir um novo doente no sistema.	36
Figura 8 – <i>Mockup</i> da listagem de todos os doentes do sistema com ocorrência de um alerta.	36
Figura 9 – <i>Mockup</i> da interface com informação do doente e indicação de alerta.	37
Figura 10 – Diagrama de componentes do sistema a implementar.	39
Figura 11 – Diagrama de atividades do sistema a implementar.	40
Figura 12 – Diagrama de sequência para o caso “inserir um novo doente no sistema”.	42
Figura 13 – Diagrama de sequência para o caso “aceder aos dados de um doente individualmente”.	42
Figura 14 – Diagrama de sequência para o caso de ocorrência de um alarme.	43
Figura 15 – Fluxo de tarefas durante o desenvolvimento do sistema.	48
Figura 16 – Aplicação <i>Android</i> desenvolvida para conexão com as pulseiras de teste. (a) Interface inicial da aplicação, com <i>Bluetooth</i> desligado. (b) Descoberta de dispositivos <i>Bluetooth</i> nas proximidades, com indicação do nome, endereço <i>Bluetooth</i> e RSSI dos dispositivos encontrados.	54
Figura 17 – Interface da primeira versão da aplicação <i>MicrosoftBandApp</i> .	56
Figura 18 – Pedido de consentimento de acesso ao sensor de frequência cardíaca da <i>Microsoft Band 2</i> .	57
Figura 19 – Indicação em tempo real da frequência cardíaca e da temperatura corporal medidas.	58
Figura 20 – Interface da aplicação com indicação de que não estão a ser adquiridos sinais pela pulseira.	59
Figura 21 – Sinais obtidos a partir de três pontos de acesso em dBm (Fonte: (He & Chan, 2016)).	61

Figura 22 – Ícones de destaque na aplicação <i>Android</i> desenvolvida. (a) Início da atividade <i>indoor</i> . (b) Permitir localização <i>indoor</i>	63
Figura 23 – Atividade principal da aplicação <i>Android</i> para a <i>Microsoft Band 2</i>	64
Figura 24 – Lista de edifícios previstos no funcionamento da aplicação <i>Android</i> para a <i>Microsoft Band 2</i>	64
Figura 25 – Planta com indicação da posição do utilizador da aplicação.	65
Figura 26 – Modelo entidade-relacionamento correspondente ao sistema a implementar.	69
Figura 27 – Modelo conceptual de dados do sistema a implementar.	71
Figura 28 – Modelo físico de dados gerado automaticamente pelo <i>software</i> a partir do modelo conceptual.	72
Figura 29 – Exemplo de definição de um objeto através da linguagem HTML.....	74
Figura 30 - Interface inicial da <i>dashboard</i>	75
Figura 31- Interface para inserir um novo doente no sistema.....	75
Figura 32 - Lista de doentes do sistema, organizados de acordo com a classificação de triagem e com indicação de alertas.	76
Figura 33 - Informações de um dos doentes da lista, com indicação de alarme na temperatura corporal.	76
Figura 34 – Tabela com as <i>migrations</i> implementadas (Fonte: (“Creating a Basic ToDo Application in <i>Laravel 5</i> – Part 1,” 2015)).	78
Figura 35 - Tabela <i>projects</i> após a definição de <i>seeds</i> (Fonte: (“Creating a Basic ToDo Application in <i>Laravel 5</i> – Part 1,” 2015)).	79
Figura 36 – Model para a tabela “ <i>doente</i> ”, com indicação dos campos da tabela e da relação com a tabela “ <i>episodio</i> ”.	80
Figura 37 – <i>Route</i> para o <i>RESTful Controller CreateController</i> , tendo como recurso o <i>model</i> “ <i>Episode</i> ”.	81
Figura 38 – Função em <i>JavaScript</i> com recurso ao <i>Ajax</i>	82
Figura 39 – Funcionamento do SOAP (Fonte: (Quaine, n.d.)).	84
Figura 40 – Interface do <i>Postman</i> , com exemplo de um pedido ao servidor e a respetiva resposta.....	86
Figura 41 – Código <i>Java</i> gerado pelo <i>Postman</i>	87
Figura 42 – Pedido ao servidor e respetiva resposta usando o método <i>getUsersLocation()</i> . ..	88
Figura 43 – Última localização do doente 150 no interior do Edifício A do IPN.	89
Figura 44 – Componentes do sistema implementado.	91
Figura 45 – Interfaces da <i>dashboard</i> com botões demarcados a vermelho e campos a azul...	93

Figura 46 – Média das classificações dadas pelos participantes a cada uma das questões do teste de usabilidade.....96

Figura 47 – Média das classificações dadas pelos participantes a cada categoria do teste de usabilidade.....97

Lista de tabelas

Tabela 1 - Limitações na medição da temperatura corporal usando vários métodos.....	12
Tabela 2 – Sistemas comerciais e patentes de medição de sinais vitais com diversas aplicações.....	17
Tabela 3 - Protocolos da norma IEEE 802.11 (Fonte: (Poole) (Harnwood, 2009)).....	21
Tabela 4 - Características de funcionamento da tecnologia <i>Zigbee</i>	23
Tabela 5 – Estudo comparativo entre as tecnologias mais utilizadas em sistemas de localização <i>indoor</i> (Fonte: (Poole)	26
Tabela 6 – Possíveis soluções de monitorização encontradas, capazes de medir os sinais vitais desejados e de conexão <i>Bluetooth</i>	44
Tabela 7 – Pulseiras utilizadas no desenvolvimento do sistema.	52
Tabela 8 – Algumas aplicações exploradas para cada uma das pulseiras utilizadas.	55
Tabela 9 – Vantagens dos protocolos de comunicação do tipo REST e do tipo SOAP.....	85
Tabela 10 – Questões abrangidas pelo modelo USE utilizadas nos testes de usabilidade realizados.....	95
Tabela 11 – Caracterização dos participantes do teste de usabilidade realizado.....	96
Tabela 12 – Soluções propostas para ultrapassar os desafios em aberto do sistema.....	101

Lista de acrónimos

API	- <i>Application Programming Interface</i>
AoA	- <i>Ângulo de Chegada</i>
ATT	- <i>Attribute Protocol</i>
BLE	- <i>Bluetooth Low Energy</i>
CSS	- <i>Cascading Style Sheets</i>
DBMS	- <i>Database Management System</i>
DECO	- <i>Defesa do Consumidor</i>
GPS	- <i>Global Positioning System</i>
HTML	- <i>Hyper Text Markup Language</i>
ID	- <i>Identity</i>
IDE	- <i>Integrated Development Environment</i>
INEM	- <i>Instituto Nacional de Emergência Médica</i>
IPN	- <i>Instituto Pedro Nunes</i>
IPS	- <i>Indoor Positioning System</i>
ISEC	- <i>Instituto Superior de Engenharia de Coimbra</i>
ISM	- <i>Industrial, Scientific and Medical</i>
LAS	- <i>Laboratório de Automática e Sistemas</i>
LED	- <i>Díodo Emissor de Luz</i>
MVP	- <i>Minimum Viable Product</i>
NFC	- <i>Near Field Communication</i>
ORM	- <i>Object-Relational Mapping</i>
PDR	- <i>Pedestrian Dead Reckoning</i>
PHP	- <i>Hypertext Preprocessor</i>
REST	- <i>Representational State Transfer</i>
RFID	- <i>Radio Frequency Identification</i>
RSSI	- <i>Received Signal Strength Indicator</i>
SD	- <i>Desvio Padrão</i>
SDK	- <i>Software Development Kit</i>
SHF	- <i>Super High Frequency</i>
SOAP	- <i>Simple Object Access Protocol</i>
ToA	- <i>Tempo de Chegada</i>
ToF	- <i>Tempo de Voo</i>
UHF	- <i>Ultra High Frequency</i>
URL	- <i>Uniform Resource Locator</i>
USE	- <i>Usefulness, Satisfaction and Ease of use</i>
UTC	- <i>Coordinated Universal Time</i>
UUID	- <i>Universally Unique Identifier</i>
UWB	- <i>Ultra-Wide Band</i>
WSDL	- <i>Web Service Definition Language</i>
XML	- <i>eXtensible Markup Language</i>

1. Introdução

O presente relatório é fruto do estágio realizado por Adriana Patrícia Monteiro Reis, no âmbito da Unidade Curricular de Projeto/Estágio do Mestrado em Instrumentação Biomédica do Instituto Superior de Engenharia de Coimbra (ISEC), realizado na empresa *Sensing Future Technologies* e em colaboração com o Laboratório de Automática e Sistemas (LAS) do Instituto Pedro Nunes (IPN).

Ao longo deste capítulo é feita uma introdução ao trabalho desenvolvido, sendo descrito o enquadramento do estágio (Secção 1.1), a motivação que deu origem ao projeto em questão (Secção 1.2), os objetivos (Secção 1.3) e uma breve referência às empresas de acolhimento (Secção 1.4). É ainda realizada uma apresentação sumária à organização do relatório, incluindo uma breve descrição do conteúdo de cada capítulo (Secção 1.5).

1.1 Enquadramento

Existem diversas causas que motivam uma ida às urgências, embora nem todas as situações sejam realmente urgentes. Na realidade, alguns dos problemas de saúde que surgem nas urgências poderiam ter sido resolvidos recorrendo aos cuidados primários. No entanto, o serviço de urgência dos hospitais continua a ser a primeira opção de grande maioria dos doentes, sobretudo porque estes acreditam que num hospital têm acesso a profissionais médicos mais especializados e meios complementares de diagnóstico melhores do que aqueles que se encontram disponíveis nos centros de saúde (Carriço, 2017).

Conjugando a grande afluência aos serviços de urgência com a insuficiência de profissionais de saúde nestes serviços leva a que, muitas vezes, o doente seja sujeito a tempos de espera que ultrapassam largamente os padrões teoricamente definidos. Assim, após o doente ser triado à chegada ao serviço de urgência, poderá ter de esperar durante muito tempo para ser visto pelo médico, tempo durante o qual a sua condição de saúde pode ser agravada sem que o profissional de saúde responsável se aperceba, o que pode motivar, em casos extremos, à ocorrência de uma fatalidade. Idealmente, para evitar este tipo de situações, o estado de saúde do doente que recorre às urgências deveria ser avaliado frequentemente após a triagem e até ao final do episódio de urgência. Mais uma vez, a falta de recursos humanos e materiais, bem como o excesso de doentes nestes serviços, faz com que a falta deste tipo de acompanhamento se torne cada vez mais numa realidade.

1.2 Motivação

Atualmente, existem vários sistemas de monitorização de sinais vitais, bem como sistemas de localização no interior de edifícios. No entanto, nenhuma destas tecnologias se encontra aplicada, de forma conjugada, a um serviço de urgência: é o profissional de saúde que é responsável por, pontualmente, se dirigir ao doente, o que pode representar uma tarefa

inviável devido ao tempo que é necessário, e verificar se os seus parâmetros vitais se encontram normalizados ou se, em caso contrário, o doente se encontra em dificuldades.

Recorrendo às tecnologias atualmente existentes, é possível o desenvolvimento de um sistema apto para uma monitorização síncrona capaz de alertar o profissional de saúde para a existência de complicações no estado de saúde do doente, melhorando assim não só o serviço prestado pelo hospital, mas também contribuir para que o doente se sinta mais acompanhado e, por conseguinte, mais tranquilo e menos ansioso. É ainda possível dar indicação da localização do doente no hospital, progredindo assim no sentido de melhorar a rapidez de atuação do profissional de saúde permitindo-lhe encontrar o doente e, conseqüentemente, melhorar o funcionamento das urgências.

1.3 Objetivos

Dado o quadro atual dos serviços de urgência portugueses e a necessidade de melhorar a qualidade do atendimento dos doentes que a eles recorrem, surgiu assim por parte da *Sensing Future Technologies* e em colaboração com o Instituto Pedro Nunes a idealização de um sistema capaz de monitorizar o doente de forma constante, permitindo assim alertar para possíveis alterações da sua condição de saúde enquanto este permanece na urgência hospitalar.

Desta forma, o objetivo deste estágio é a conceção de um sistema de monitorização a ser usado em ambiente hospitalar, capaz de medir os sinais vitais da frequência cardíaca e da temperatura corporal, assim como lançar um alerta no caso de os valores medidos se encontrarem fora dos padrões normais. Deverá ainda dar informação acerca da localização do doente para que o profissional de saúde o possa localizar rapidamente dentro do hospital, e assim dar-lhe a assistência necessária no menor intervalo de tempo possível. Todas estas informações deverão ser possíveis de visualizar recorrendo a uma *dashboard* intuitiva.

Para atingir os objetivos do estágio foram desenvolvidas um conjunto de atividades, cada uma delas composta por tarefas, e que englobam o processo de pesquisa, planeamento e implementação do protótipo que constitui o objetivo principal do estágio.

Assim, a primeira atividade consistiu numa análise ao estado da arte, incidindo principalmente na pesquisa de soluções de monitorização de processos vitais e na pesquisa sobre processos de triagem.

Por sua vez, a segunda atividade recaiu sobre o problema e o levantamento de requisitos para a solução a desenvolver.

A terceira atividade pretendeu definir a arquitetura do sistema através da pesquisa e estudo de ferramentas a utilizar e a identificação de linguagem de programação de acordo com os requisitos identificados, resultando na apresentação da arquitetura global do sistema.

De seguida, a quarta atividade incidiu no desenvolvimento e implementação da solução a partir do desenvolvimento de um sistema de aquisição de parâmetros vitais, de um sistema de localização *indoor* e de uma *dashboard* de monitorização.

A quinta atividade recaiu sobre a realização de um teste do sistema, sendo necessário definir quais os testes a realizar para a verificação do seu funcionamento.

Por fim, procedeu-se à implementação do protótipo em ambiente piloto, instalando-se a solução desenvolvida e recolhendo o *feedback* e a avaliação do sistema.

1.4 Empresa de acolhimento

A empresa de acolhimento do estágio realizado foi a *Sensing Future Technologies*, em colaboração com o Laboratório de Automática e Sistemas do Instituto Pedro Nunes.

A *Sensing Future Technologies* é uma empresa vocacionada para conceber, desenvolver e implementar dispositivos médicos tecnológicos inovadores e flexíveis, dirigidos a várias aplicações na área da saúde. Constituída em 2011 e com sede na Incubadora de Empresas do Instituto Pedro Nunes, é constituída por uma equipa multidisciplinar, com destaque nas áreas de engenharia mecânica, biomecânica, eletrotécnica e informática, e com experiência em instrumentação, controlo de sistemas e robótica (*Sensing Future Technologies*, n.d.).

O projeto é ainda coorientado pelo Laboratório de Automática e Sistemas do Instituto Pedro Nunes, que em colaboração com empresas concebem novas tecnologias no domínio da eletrónica, automação, robótica e instrumentação. As principais áreas em que incidem os projetos desenvolvidos no Laboratório de Automática e Sistemas são a *Ambient Assisted Living*, saúde e qualidade de vida, mobilidade de pessoas e bens e agricultura de precisão, instrumentação e monitorização (Instituto Pedro Nunes, n.d.).

1.5 Organização do relatório

O presente relatório encontra-se dividido em seis capítulos, cada um constituído por secções. Dá-se importância a uma sequência lógica da evolução do texto, estando assim os capítulos organizados de forma a facilitar a compreensão do trabalho desenvolvido.

No presente capítulo foi dada uma visão geral sobre o projeto desenvolvido, nomeadamente, dando-se destaque ao enquadramento, à motivação e aos objetivos do estágio. São ainda apresentadas, de uma forma breve, a instituição de acolhimento do estágio e o laboratório que colaborou na orientação do mesmo.

O Capítulo 2 compreende uma análise do estado da arte. São abordados temas como a caracterização dos serviços de urgência atuais (nacionais e internacionais), a descrição do protocolo usado na classificação de prioridades no atendimento, os quatro sinais vitais que tipicamente são alvo de monitorização, de que forma podem estes ser monitorizados e qual a

sua importância na verificação do estado de saúde de um doente. Por fim é feito um levantamento sobre os métodos e tecnologias que podem ser aplicáveis a sistemas de localização *indoor*.

No Capítulo 3 são definidos quais os requisitos necessários ao funcionamento do sistema desenvolvido, partindo das funcionalidades desejadas que este deve oferecer. São definidas as especificações técnicas da *dashboard* e da arquitetura do sistema, bem como as características desejáveis para o *hardware* de monitorização de sinais vitais. Para além de um levantamento e análise dos dispositivos atualmente existentes no mercado, são também apresentadas as *user stories* do sistema, os *mockups* executados e diagramas de componentes, atividades e sequência que permitem explicar o seu funcionamento.

No Capítulo 4 é explicado todo o desenvolvimento do projeto, desde a escolha do *software* e *hardware*, a explicação sobre o desenvolvimento de uma aplicação *Android* para a recolha de sinais vitais com capacidade de localização do utilizador e a descrição da *dashboard* que permite ver toda a informação necessária relativa aos doentes no decorrer de um episódio de urgência.

Do Capítulo 5 consta a descrição do protótipo desenvolvido, abrangendo os seus componentes, qual a função de cada um no sistema, e o modo de utilização do protótipo. Foram realizadas avaliações do sistema, bem como testes de usabilidade que serão descritos ainda neste capítulo.

Por fim, no Capítulo 6, são apresentadas as conclusões relativamente ao trabalho desenvolvido e ao estágio realizado, bem como algumas sugestões de trabalho futuro que poderão acrescentar valor ao protótipo desenvolvido.

2. Estado da arte

Nos últimos anos tem sido assunto recorrente nos meios de comunicação social a insuficiência de recursos (humanos e materiais) nos serviços de urgência portugueses, o que pode levar à ocorrência de situações fatais para doentes de risco devido ao tempo de espera entre a triagem e uma nova observação. Atendendo aos objetivos deste projeto, considerou-se necessário um breve estudo prévio sobre algumas temáticas, tais como serviços de urgência, medição de sinais vitais e a localização *indoor*.

Na Secção 2.1 são abordados os serviços de urgência, começando pelo seu funcionamento e pelo sistema de triagem atualmente mais comum nos hospitais de todo o mundo. Relativamente aos serviços de urgência em Portugal, pretende-se dar uma visão geral sobre qual o seu estado atual, e quais as medidas tomadas para tentar desviar os doentes, com condição de saúde pouco urgente ou não urgente, da sua utilização. Devido aos tempos de espera significativos a que o doente pode estar sujeito enquanto espera por ser observado pelo médico após a triagem, a sua condição de saúde pode piorar sem que seja notada pelos profissionais de saúde, resultando por vezes em fatalidades, pelo que será também abordada esta temática.

De seguida, a Secção 2.2 incide em quatro sinais vitais, sendo eles a temperatura corporal, a frequência cardíaca, a taxa respiratória e a saturação do oxigénio no sangue, e métodos através dos quais podem ser medidos, pretendendo-se ainda salientar a importância da sua medição na triagem e o impacto na alteração da condição de saúde dos doentes. Através dos métodos de medição apresentados será possível justificar a escolha dos parâmetros que se deseja medir neste sistema, uma vez que nem todos os métodos oferecem uma solução possível de integrar num só dispositivo, pequeno e confortável de usar. Por fim, são apresentados alguns sistemas comerciais ou patentes de dispositivos de monitorização de sinais vitais com diversas aplicações.

Na Secção 2.3, para auxiliar na argumentação da escolha do sistema de localização *indoor* a utilizar no projeto, e que é crucial para conhecer a localização de um doente no hospital, são abordados métodos e tecnologias que podem atualmente ser aplicáveis, fazendo-se ainda uma comparação entre cada um deles e as possíveis associações. São ainda apresentados alguns sistemas comerciais e patentes de sistemas de localização *indoor*.

Por fim, na Secção 2.4, são expostas algumas considerações finais acerca do estado da arte, pretendendo-se salientar a importância do protótipo desenvolvido para uso em ambiente de urgência hospitalar, bem como justificar algumas escolhas relativas aos sinais vitais que são medidos e aos métodos e tecnologias usados pelo sistema de localização *indoor* adotado na solução final.

2.1 Serviços de urgência

Um serviço de urgência, tipicamente, é frequentado por doentes com condições de saúde de diferentes níveis de gravidade; em alguns dos casos, o doente tem de ser atendido de imediato

ou dentro de poucos minutos e, em outros casos, o doente poderá esperar por um período de tempo mais longo sem que a sua vida seja posta em causa (“Hospitais a rebentar pelas costuras”, 2014).

As razões que levam um doente a dirigir-se a um serviço de urgência são bastante variadas e incluem desde acidentes de automóvel a quedas, queimaduras, ataques cardíacos, dificuldades respiratórias associadas a doenças como asma ou pneumonias, febres altas, dores persistentes, reações alérgicas, entre outras (Carl Bianco, 2000). Torna-se, assim, necessário que num serviço de urgência se possam estabelecer de forma eficiente prioridades no atendimento dependendo da condição de saúde de cada um dos doentes.

2.1.1 Funcionamento típico de um serviço de urgência

Aquando da entrada de um doente nas urgências, o primeiro local onde este se deverá dirigir é a área de admissão e receção, onde são realizados todos os processos administrativos. Após esta etapa, é necessário definir qual a sua prioridade de atendimento, pelo que o doente é encaminhado para uma sala de triagem, para que possa ser observado e conduzido para a especialidade médica mais adequada. Este passo pode não existir caso o doente chegue ao hospital em risco de vida e tenha de ser encaminhado imediatamente para a sala de emergência. Numa situação em que o doente não corre risco de vida, deverá de seguida dirigir-se à área apontada na triagem, tendo de aguardar para ser observado pelo médico, ou pode ainda ser necessária a realização de alguns exames médicos (Gomes, 2013).

Após a observação do doente, e eventualmente do resultado dos exames, o médico irá determinar se o doente pode regressar a casa, tipicamente acompanhado de recomendações e de uma receita médica, ou se o doente necessita de cuidados de saúde no hospital, e portanto determina o seu internamento numa enfermaria (Kelly, 2014).

De forma a definir qual a prioridade de atendimento de um doente, é feita uma triagem, em que o protocolo de triagem tipicamente adotado é o Protocolo de Manchester, que classifica cada situação como emergente (a ser correspondida imediatamente), urgente, sem necessitar de tratamento imediato, ou menos urgente (Carl Bianco, 2000).

2.1.2 Triagem de Manchester

Em novembro de 1994 foi criado o Grupo de Triagem de Manchester, constituído por médicos e enfermeiros dos vários Serviços de Urgência de Manchester, desde serviços de urgência gerais a serviços de urgência pediátricos e oftalmológicos. O objetivo deste grupo era implementar um sistema de triagem comum a todos os serviços de urgência que fosse consistente e que tivesse em vista, acima de tudo, a segurança dos doentes que os frequentam.

Em 1997 foi publicada a primeira edição de “*Emergency Triage*” pela BMJ Books, e cerca de um ano depois foi traduzido para outras línguas e publicado em países como Áustria, Brasil e Alemanha, e também em Portugal. O Protocolo de Manchester é, assim, o sistema

mais comumente adotado em todo o mundo e usado na triagem de milhões de doentes por ano (“Portugal - Manchester Triage System”).

Este sistema de triagem utiliza um código de cores, no qual cada cor corresponde às situações emergente, muito urgente, urgente, pouco urgente ou não urgente. Estas são atribuídas de acordo com o fluxograma representado na Figura 1 (Gomes, 2013).

Para responder às questões do fluxograma é importante medir alguns sinais vitais do doente, em especial a temperatura, a pulsação, a taxa respiratória e a pressão sanguínea, bem como conhecer outros dados como problemas de saúde anteriores, medicação que habitualmente toma e alergias (Carl Bianco, 2000).

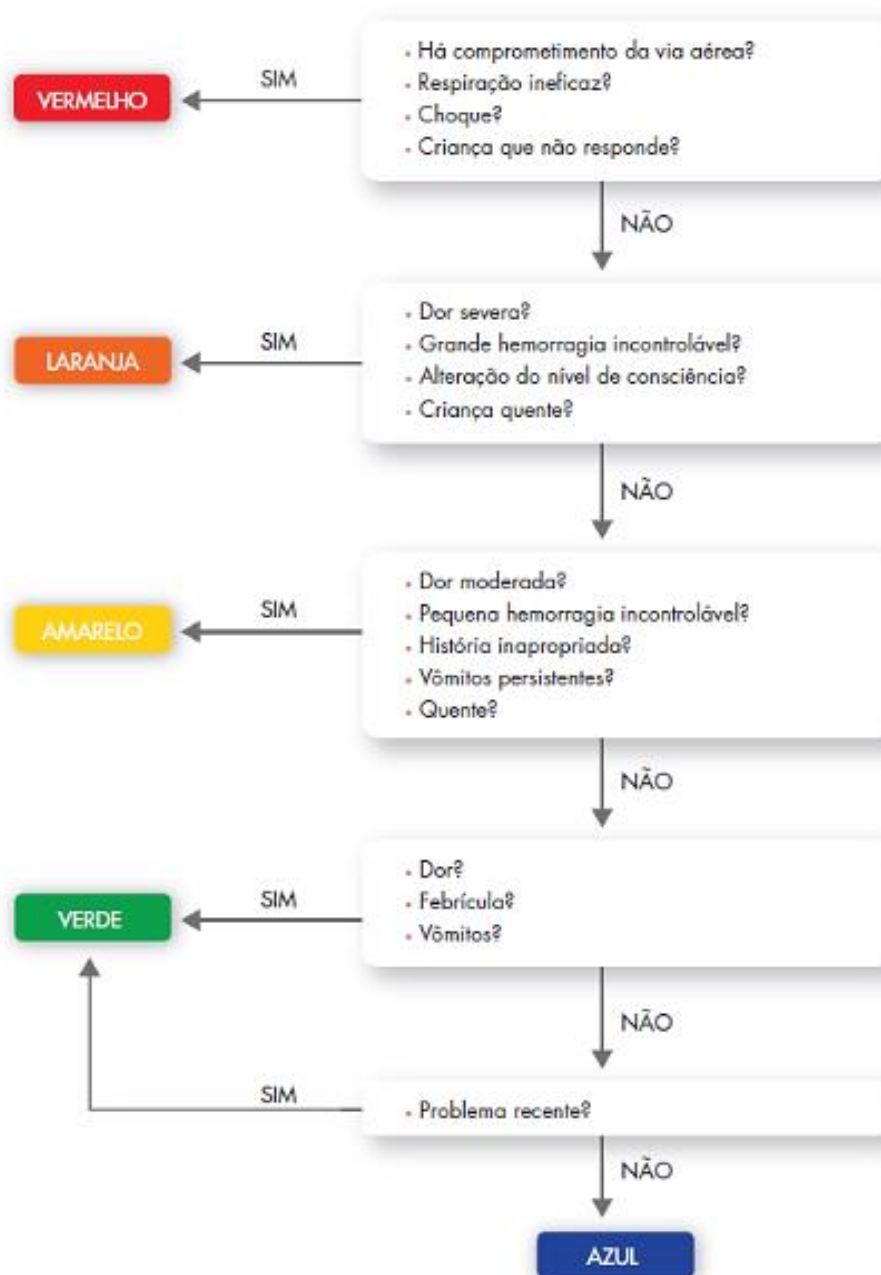


Figura 1 - Atribuição de cores das pulseiras na Triagem de Manchester (Fonte:(Gomes, 2013)).

A cada cor está ainda associado um tempo de espera até ao atendimento pelo médico da especialidade, como mostra a Figura 2. Na prática, este tempo de espera poderá ser mais curto ou mais longo, dependendo de fatores como o fluxo de doentes no serviço de urgência, sendo completamente imprevisível no caso de surgirem ocorrências inesperadas.



Figura 2 - Tempos de espera de acordo com a Triagem de Manchester (Fonte: (“Hospitais a rebentar pelas costuras”, 2014)).

Um exemplo destas ocorrências é a elevada afluência de doentes na época típica de gripes, pelo que estes tempos de espera nunca são garantidos e o tempo de visita às urgências poderá ser longo (Kelly, 2014). Este facto revela a importância de um sistema de monitorização contínuo em ambiente de urgência hospitalar de forma a melhorar a qualidade dos serviços de urgência e a evitar situações de grave degradação de saúde dos doentes.

2.1.3 Serviços de urgência em Portugal

Em Portugal, o sistema de triagem utilizado desde o ano 2000 é o Protocolo de Manchester, tendo sido adotado primeiramente em Lisboa e no Porto, e estando atualmente a ser usado em cerca de 81 serviços de urgência em todo o país (“Portugal - Manchester Triage System”).

De acordo com este sistema e segundo um estudo levado a cabo pela Defesa do Consumidor (DECO), estima-se que 4 em cada 10 utentes num serviço de urgência não necessitam de atendimento imediato, e apenas 13% estariam em situação crítica (“Hospitais a rebentar pelas costuras”, 2014). As causas apontadas para este comportamento são a proximidade ou a facilidade de acesso que este tipo de serviços pode oferecer relativamente aos centros de saúde, para além de meios complementares de diagnóstico que muitas vezes não existem nos cuidados primários, pelo que o doente que recorre a estes pode ser encaminhado posteriormente para um hospital. Em outros casos ainda, poderá existir uma grande dificuldade de marcação de consultas em tempo útil por falta de médico de família, uma realidade que afeta uma grande parte da população portuguesa (Gomes, 2013).

Com vista a uma melhor utilização do sistema nacional de saúde por parte da população, que poderia evitar o recurso aos serviços de urgência dos hospitais em caso de condições de saúde não urgentes em que o doente pode recorrer a outros meios, foram criadas várias medidas e recursos.

Existe em Portugal a Linha Saúde 24, que se destina a ajudar pessoas com problemas de saúde supostamente menos graves e a reencaminhar os utentes para os serviços adequados à sua condição. No entanto, existe ainda uma escassa utilização desta linha, por falta de conhecimento ou hábito, o que é um fator que contribui para o excesso de utentes nas urgências em situações pouco urgentes ou não urgentes (“Hospitais a rebentar pelas costuras,” 2014).

Foram ainda aumentadas as taxas moderadoras, sendo mais altas nos serviços de urgência de hospitais e mais baixas em centros de saúde, com o objetivo de haver uma maior procura destes últimos por parte da população (Gomes, 2013). Esta medida permitiu, segundo um estudo da Universidade Nova de Lisboa, que parte da população não só não tenha recorrido às urgências, como não tenha recorrido ao centro de saúde para consultas com o médico de família. Para além disso, algumas das pessoas inquiridas não realizaram determinados exames de diagnóstico nem tomaram certos medicamentos prescritos devido aos custos associados (“Índice de Saúde Sustentável,” 2016).

No dia 1 de fevereiro de 2016 foi também lançada uma aplicação pelo Serviço Nacional de Saúde que permite consultar o tempo de espera real das urgências de vários hospitais e centros de saúde portugueses, bem como quanto tempo irá demorar até a marcação de uma consulta de determinada especialidade (Borges, 2016). O objetivo é evitar picos de procura nos serviços de urgência, permitindo ao utilizador decidir qual a melhor opção para a sua situação de saúde em determinado momento. A aplicação apenas apresenta dados recebidos pela instituição, sendo esta responsável pelo envio da informação (“MySNS Tempos - A Nova Versão da App Já Disponível”). Em novembro de 2016 esta aplicação foi redesenhada e foi lançada uma nova versão, com o objetivo de ser mais rápida e apresentar mais e melhor informação. Para além disso, tem ainda um código de identificação de cores para daltónicos, podendo assim atingir um maior número de pessoas.

Em abril de 2016 passam a estar isentos de pagamento de taxas moderadoras nos serviços de urgência e em exames de diagnóstico utentes referenciados pelos centros de saúde e pelo Instituto Nacional de Emergência Médica (INEM) (“Utentes referenciados por centros de saúde ou Saúde 24 têm prioridade,” 2016). Em instituições hospitalares integradas no Serviço Nacional de Saúde, é ainda por lei dado atendimento prioritário a utentes referenciados através dos centros de saúde ou da Linha Saúde 24 (“Utentes referenciados por centros de saúde ou Saúde 24 têm prioridade,” 2016).

2.1.4 Mortalidade nos serviços de urgência portugueses

A mortalidade nas urgências portuguesas é cada vez mais uma realidade. De acordo com a revista *Visão*, na edição de Janeiro de 2015 refere-se que houve um aumento de 25% da mortalidade face ao mesmo período do ano anterior (*Visão*, 2015).

Segundo a afirmação do Bastonário da Ordem dos Médicos, José Manuel Silva, "só não foram noticiadas mais mortes, resultantes da assistência indevida, porque as famílias, provavelmente, não se aperceberam", apontando como motivo o atraso no atendimento a doentes que acabam por falecer nas urgências sem que as famílias associem o falecimento ao

mau funcionamento do serviço (Expresso, 2015). Um dos casos reporta a triagem de um doente com pulseira amarela, a quem foi mais tarde atribuída uma pulseira laranja devido ao agravamento do estado de saúde, tendo o doente falecido após cerca de seis horas de espera no serviço de urgência (Cm, 2015); recorde-se que o tempo de atendimento a um doente com pulseira amarela é de uma hora, em concordância com a Figura 2.

De acordo com os profissionais de saúde entrevistados para o artigo da Visão, este aumento de mortes nas urgências portuguesas deve-se à falta de médicos, de enfermeiros, de assistentes operacionais e de recursos, a cortes orçamentais, à emigração ou reforma de profissionais sem que outro seja colocado no seu lugar, e a diminuição da capacidade de internamento dos doentes nos hospitais devido à falta de materiais e recursos. O seguinte excerto caracteriza o sistema nacional de saúde em 2015, indo de encontro às últimas notícias de mortes nas urgências que ainda em 2017 tiveram destaque na comunicação social: “Com a crise dos últimos anos, tudo se complicou: os doentes chegam em pior estado. Muitas vezes, atrasam a ida à urgência. Ou então não tomam os medicamentos como deviam e em caso de doenças crónicas descompensam mais depressa. Com menos pessoal, o cenário é mais do que desolador, com corredores e salas cheias de gente que devia estar em observação mas acaba por ficar por ali, sem qualquer vigilância” (Visão, 2015).

2.2 Sistemas de monitorização de sinais vitais

A vigilância dos doentes permite detetar mudanças na sua condição de saúde, conseguindo assim reconhecer uma deterioração clínica atempadamente (Elliott & Coventry, 2012). De acordo com Schmidt, alguns casos de mortalidade em hospitais poderiam ser evitados se a monitorização dos sinais vitais do doente fosse eficiente, permitindo assim um reconhecimento precoce de situações críticas que podem conduzir à morte (Schmidt et al., 2015).

Esta monitorização é feita, tipicamente, de acordo com quatro sinais vitais, sendo eles a temperatura corporal, a frequência cardíaca, a taxa respiratória e a saturação do oxigénio no sangue (Elliott & Coventry, 2012). De acordo com o fluxograma de atribuição de cores na Triagem de Manchester (Figura 1), a temperatura corporal é um parâmetro bastante importante, bem como fatores como o nível de consciência e a dor. Nesta secção serão analisados os quatro sinais vitais e a forma como poderão ser mensurados (Subsecção 2.2.1), salientando-se a importância da sua monitorização nos serviços de urgência (Subsecção 2.2.2). Na Subsecção 2.2.3 serão apresentados alguns sistemas de medição de sinais vitais utilizados em diversas aplicações e com diferentes características.

2.2.1 Métodos de medição de sinais vitais

Os sinais vitais são simples medidas de parâmetros fisiológicos do corpo humano, que incluem, tipicamente, a temperatura corporal, a frequência cardíaca, a taxa respiratória e a saturação do oxigénio no sangue. São indicadores variáveis e dinâmicos, e que podem ser afetados por diversos fatores, dos quais fazem parte a medicação e as drogas recreativas, pelo

que, por si só, não conseguem dar uma informação exata da condição de saúde de um doente, tendo no entanto um papel importante na tomada de decisões quando fora dos valores padrão (Gilboy, Tanabe, Travers, Rosenau, & Eitel, 2005). Para entender a importância dos sinais vitais, torna-se assim essencial saber o que são e como podem ser avaliados.

2.2.1.1 Temperatura corporal

A temperatura corporal representa o balanço entre a produção e a perda de calor, denominada de termorregulação. O valor padrão para este sinal vital é de aproximadamente 36,5 °C (Elliott & Coventry, 2012).

Existem ainda vários métodos de medição da temperatura corporal, em que os valores podem apresentar diferenças entre si de acordo com a zona corporal em que é feita a medição, bem como pela calibração do dispositivo de medição. Pode, assim, medir-se a temperatura na cavidade oral, no reto, na axila, na orelha (temperatura do tímpano), ou à superfície da pele.

Nos primeiros três métodos de medição pode usar-se o mesmo dispositivo: um termómetro de vidro, que tipicamente usa uma coluna de mercúrio, ou um termómetro digital, em que a medição é feita de forma eletrónica. No entanto, comparando os valores de temperatura obtidos em cada uma destas partes do corpo, verificam-se diferenças: a temperatura retal tende a ser entre 0,5 ou 0,7 °F (cerca de 0,28 a 0,39 °C) mais elevada relativamente à temperatura medida na cavidade oral, e por sua vez a temperatura axilar tende a ser 0,3 ou 0,4 °F (cerca de 0,17 a 0,22 °C) inferior a esta (John Hopkins Medicine). Já no caso da medição de temperatura do tímpano, é usado um termómetro de infravermelhos, que mede o calor radiante emitido pela membrana do tímpano (Childs, Harrison, & Hodkinson, 1999), e para a medição da temperatura superficial é usado um termómetro especial tipicamente colocado na testa (John Hopkins Medicine). Estes termómetros são capazes de medir o calor naturalmente emitido pela artéria temporal, na zona do infravermelho (Allegaert, Casteels, Gorp, & Bogaert, 2014). A Figura 3 representa os dois últimos tipos de termómetros mencionados.



Figura 3 - Tipos de termómetros. (a) Termómetro usado para medir a temperatura do tímpano. (b) Termómetro para medir a temperatura superficial da pele (Fonte: *Google Imagens*).

Apesar de todos estes métodos serem não invasivos e de permitirem uma medição rápida e fácil, todos apresentam limitações e fatores que afetam a medição. Estes fatores encontram-se descritos na Tabela 1.

Tabela 1 - Limitações na medição da temperatura corporal usando vários métodos.

Método de Medição	Limitações
Temperatura Oral	Pode ser afetada pelo consumo de bebidas quentes ou frias (Marquez & Ballen, 1945); a taxa respiratória pode também afetar a medição (Bernhardt, 2008).
Temperatura Rectal	Risco de perfuração rectal no caso de bebés ou jovens, peritonite ou quebra do termómetro durante a medição (Marquez & Ballen, 1945).
Temperatura Axilar	A medição é afetada pela temperatura ambiente (Bernhardt, 2008).
Temperatura do Tímpano	É necessário que a ponta de prova consiga “ver” a membrana do tímpano, caso contrário a medição obtida difere bastante da temperatura real; no caso da medição em crianças há influência da temperatura ambiente, dado o canal auditivo ser mais curto (Childs et al., 1999); a presença de cera no ouvido afeta a medição (Marquez & Ballen, 1945).
Temperatura Superficial da Pele	A medição é afetada pela temperatura ambiente (Bernhardt, 2008).

Na pediatria, principalmente, o método que melhor traduz a temperatura corporal é a temperatura retal; no entanto, devido aos riscos que esta medição acarreta, uma boa alternativa será usar a temperatura do tímpano, ou efetuar a medição da temperatura superficial da pele (Allegaert et al., 2014). Na generalidade dos casos, a medição da temperatura axilar é a mais usada por ser a mais prática (“Termómetros: dicas para escolher e medir a temperatura,” 2015).

2.2.1.2 Frequência cardíaca

Neste ponto torna-se importante fazer a distinção entre frequência cardíaca e pulsação. A frequência cardíaca é definida como o número de vezes que o coração se contrai num minuto (“Difference Between Heart Rate and Pulse”). Pode, assim, ser traduzida em batimentos por minuto, sendo que o normal em adultos saudáveis é uma frequência cardíaca entre 60 e 100 batimentos por minuto (John Hopkins Medicine, n.d.). Por sua vez, a pulsação traduz-se na variação do fluxo de sangue que circula através dos capilares sanguíneos, causado pelos batimentos do coração (“Difference Between Heart Rate and Pulse”). A medição da pulsação inclui não só a frequência cardíaca mas também o ritmo cardíaco e a força do impulso. Este sinal vital é afetado por diversos fatores, sendo eles a idade, a condição de saúde, a medicação, a atividade e as emoções. É, por exemplo, normal que um atleta tenha uma frequência cardíaca de 40 batimentos por minuto, sem que isso signifique que tenha uma

condição especial; apenas tem uma atividade cardiovascular mais intensa (John Hopkins Medicine). Os métodos de medição da frequência cardíaca incluem eletrocardiografia, fotopletismografia, oscilometria e fonocardiografia.

A eletrocardiografia é um método de medição que requer a colocação de dois ou mais elétrodos em pontos específicos do corpo humano, de forma a conseguir detetar-se a variação do impulso elétrico gerado durante o ciclo cardíaco pelo nodo sinoatrial. O eletrocardiograma é constituído por seis picos e consequentes vales, que permitem obter uma média da frequência cardíaca. A partir da forma de onda do eletrocardiograma pode calcular-se a frequência cardíaca através da distância entre o pico R do complexo QRS de dois ciclos consecutivos, e obter uma média num período de tipicamente 15, 30 ou 60 segundos.

Usando a fotopletismografia é detetada a variação do volume de sangue que ocorre numa veia em cada ciclo cardíaco (Sanjeev Kumar, n.d.). O dispositivo de medida inclui um Díodo Emissor de Luz (LED) infravermelho de baixa intensidade, capaz de atravessar tecidos biológicos, sendo absorvido em parte pelos ossos, pigmentos da pele e pelo sangue, tanto venoso como arterial. Como tal, a variação na intensidade da luz depende do fluxo de sangue, e pode ser captada por um sensor fotossensível, como um fotodíodo. Este tipo de tecnologia pode tornar-se bastante barata, pequena, robusta e fácil de utilizar. Permite ainda ser usada na medição de outros parâmetros, como é o caso da saturação do oxigénio no sangue (Cheriyedath, 2016).

No caso da oscilometria é detetada a pulsação das veias devido à contração e relaxamento do coração e da variação do fluxo de sangue através de sensores piezoelétricos dispostos em locais específicos, como no pulso. Estes sensores detetam a variação da pressão, e a frequência cardíaca corresponde à distância entre dois picos.

Por último, a fonocardiografia tem como objetivo captar o som da abertura e fecho das válvulas do coração produzido no bombeamento do sangue, e que pode ser facilmente detetado através do estetoscópio. Um microfone será capaz de captar este som, bem como outros sons anormais do coração, e filtrar os sons com diferentes características, permitindo assim obter um gráfico base para o cálculo da frequência cardíaca (Sanjeev Kumar, n.d.).

2.2.1.3 Taxa respiratória

A ventilação normal é o ato de inspirar e expirar acompanhado da extensão e contração da caixa torácica. É um sistema automático, que normalmente não exige esforço, e obedece a um ritmo aproximadamente constante, com um volume inspiratório também constante, pelo que as anomalias ocorrem no ritmo respiratório, na frequência e no esforço para respirar (Braun, 1990). A taxa respiratória é, assim, o número de respirações por minuto, e é medida em descanso, contando o número de inspirações durante um minuto. Para um adulto saudável, a taxa respiratória deve ser entre 12 a 16 inspirações por minuto (John Hopkins Medicine).

Os métodos de medição da taxa respiratória são bastante diversificados, incluindo: monitorização por contacto através da deteção de movimento, da monitorização do fluxo de ar, de métodos acústicos, ou do uso de eletrocardiografia e da fotopletismografia;

monitorização que não exige contacto, através da termografia por infravermelhos, e mais recentemente do uso de aplicações para *smartphones* que permitem estimar este valor.

O método de contacto visa várias medições, como a variação dos movimentos abdominais e do peito durante o ciclo respiratório, a análise do fluxo de ar e da variação da vibração durante o ciclo respiratório, a quantidade de dióxido de carbono exalada e o cálculo da taxa respiratória através do eletrocardiograma ou da saturação do oxigénio no sangue.

A deteção de movimento pode ser feita através de bandas colocadas no peito capazes de medir a alteração na impedância torácica durante o ciclo respiratório. Tipicamente é usada uma coluna de mercúrio ou métodos de variação da impedância. Este método é ainda usado para a monitorização da respiração durante o sono, pois permite uma monitorização contínua. Por sua vez, a monitorização do fluxo de ar pode ser feita através de termístores colocados no nariz do doente e que permitam detetar variações na temperatura do ar, através de transdutores capazes de medir o volume de ar exalado e de outros sensores que efetuem a medição de dióxido de carbono expirado. Um outro método utiliza o eletrocardiograma, também usado para a medição da frequência cardíaca, e efetua a medição através da flutuação do eletrocardiograma que é afetada pela taxa respiratória. Tal como este método, a fotopletiografia permite usar a forma de onda obtida para a deteção da frequência cardíaca para este cálculo.

A monitorização da taxa respiratória por métodos que não exigem contacto com o corpo do doente é menos invasiva, pelo que estes métodos são preferíveis aos anteriores no caso de crianças. Nestes métodos inclui-se a termografia por infravermelhos, que monitoriza a temperatura da face do doente que aumenta durante a expiração, permitindo assim calcular a taxa respiratória. Atualmente, as funcionalidades dos *smartphones* permitem ainda várias aplicações, nomeadamente de saúde, pelo que existem aplicações para calcular a taxa respiratória. Estas aplicações podem utilizar a câmara incorporada no *smartphone* para detetar a variação da ruborização da face e o movimento do peito durante a respiração; podem ainda determinar o tempo entre duas inspirações através da colocação do dedo no ecrã tátil (Daw et al., 2016).

2.2.1.4 Saturação do oxigénio no sangue

A saturação do oxigénio no sangue indica a percentagem de moléculas de hemoglobina presentes no sangue arterial, variando entre 0 e 100%. Em adultos saudáveis, os valores normais são de entre 94% a 100% (Smith, 1992).

Os glóbulos vermelhos, constituintes do sangue, contêm hemoglobina, que por sua vez é capaz de transportar quatro moléculas de oxigénio no estado de saturação. Se todas as moléculas de hemoglobina estiverem saturadas, então a saturação de oxigénio no sangue é de 100%. No entanto, existem alguns fatores que afetam este valor, como a altitude. Outros fatores ainda são uma má respiração, uma via aérea obstruída, uma falha no sistema circulatório e o uso de drogas que alterem as funções enunciadas.

Um valor de saturação de 90% é já considerado uma emergência clínica, e deve ser tratado de imediato. O efeito visível de uma baixa saturação é a mudança de cor da língua e dos lábios do doente, passando de um tom rosado a um tom azulado, o que pode tornar-se difícil de detetar num doente com um tom de pele escuro (World Health Organization, 2011). Assim, a saturação do oxigénio no sangue é um indicador importante do estado cardiorrespiratório de um doente, e é bastante usado em cuidados de saúde, incluindo nos serviços de urgência (Haymond, 2006).

A medição da saturação do oxigénio no sangue é executada usando a quantidade de sangue arterial, isto é, sangue que transporta oxigénio. De facto, a quantidade de sangue arterial modifica-se durante curtos períodos de tempo devido à pulsação, ainda que uma quantidade mensurável permaneça constante. Deste modo, a quantidade de sangue oxigenado pode ser separada de outros componentes (Smith, 1992).

Existem dois métodos bastante usados de medição da saturação do oxigénio no sangue, sendo estes a gasometria arterial e a oximetria de pulso. A gasometria arterial consiste numa punção vascular direta na artéria radial, por ser a mais acessível, cujo objetivo é obter informação acerca do equilíbrio ácido-base num ponto específico, e assim concluir acerca da eficácia da ventilação e da oxigenação através da quantidade de CO₂ existente nesse ponto. É bastante utilizado em internamentos por ser um método rápido, e permite ajustar o tratamento em caso de doenças graves determinando as trocas gasosas devidas à respiração e às atividades metabólica e renal (Danckers, 2016). Por sua vez, a oximetria de pulso é a medição da transmissão de luz através de uma zona do corpo bastante irrigada, como a ponta do dedo ou o lóbulo da orelha. Esta medição é feita através de um oxímetro de pulso (Haymond, 2006), constituído por dois LEDs e um fotodetector (World Health Organization, 2011). Um dos LEDs emite luz com comprimento de onda na zona do vermelho, maioritariamente absorvida pela hemoglobina desoxigenada, e o outro na zona do infravermelho, absorvida pela hemoglobina oxigenada. Parte da luz emitida é absorvida pelos tecidos, ossos e pelo sangue venoso; no entanto, esta quantidade é constante, pelo que apenas há variação na quantidade de luz vermelha e infravermelha que chega ao recetor. Usando estas duas variáveis, é assim possível calcular a saturação do oxigénio no sangue (Smith, 1992). A Figura 4 apresenta um oxímetro de pulso colocado no dedo do doente. É de notar que este oxímetro permite visualizar, para além da saturação do oxigénio no sangue (em percentagem), a pulsação em batimentos por minuto.



Figura 4 - Oxímetro de pulso (Fonte: Google Imagens).

É ainda importante referir que existem fatores que afetam a medição quando se usa um oxímetro de pulso, sendo eles a luz, que ao ser muito brilhante pode interferir na leitura, o movimento do doente, o volume de sangue, sendo difícil a medição quando há uma pulsação fraca, a vasoconstrição pelo mesmo motivo, e o envenenamento por monóxido de carbono, que cria uma falsa leitura (World Health Organization, 2011).

2.2.2 Importância dos sinais vitais na triagem

Os sinais vitais são parte integrante dos cuidados de saúde, sendo usados para diagnóstico, para detetar a deterioração da situação clínica de um doente e na triagem, de forma a assegurar que o doente com um estado de saúde mais degradado é atendido prioritariamente. Existem vários estudos realizados com o intuito de associar a mortalidade nos serviços de urgência aos sinais vitais do doente na triagem, sendo que os fatores mais evidentes de associação são a idade, o nível de consciência e a saturação do oxigénio no sangue (Ljunggren, Castrén, Nordberg, & Kurland, 2016). Para além deste facto, sinais vitais anormais podem facilmente ser associados ao internamento do doente, estando ainda associados a 98% das mortes inesperadas em serviços de urgência (Chalari et al., 2012). Valores de temperatura corporal elevados podem indicar febre ou a presença de uma infeção, enquanto que se forem baixos podem indicar um choque ou uma alteração no sistema circulatório que faça com que haja um reduzido fluxo de sangue a circular no corpo; da mesma forma, a frequência cardíaca é um indicador de alterações físicas ou mentais anormais. Por sua vez, a taxa respiratória permite detetar anomalias como apneia do sono, asma, doenças pulmonares crónicas ou anemia (Khan, Ostfeld, Lochner, Pierre, & Arias, 2016).

Os resultados demonstram que a frequência cardíaca, em conjunto com a pressão sistólica, são preditores do estado de choque do doente e permitem ainda determinar uma disfunção cardiovascular. Já a saturação do oxigénio no sangue, quando apresenta um valor baixo, pode indicar alterações no estado mental do doente ou letargia (Chalari et al., 2012). São ainda os sinais vitais ligados ao aparelho respiratório que resultam num maior número de mortalidade quando sofrem desvios, bem como aqueles ligados à atividade cardíaca e à circulação sanguínea. A frequência cardíaca aplica-se às duas situações.

Em suma, a mortalidade nos serviços de urgência pode ser associada a alterações nos sinais vitais do doente: a diminuição da saturação do oxigénio no sangue, o aumento ou diminuição da taxa respiratória e da frequência cardíaca e a diminuição da temperatura corporal, conferindo assim aos sinais vitais uma elevada importância no momento da triagem (Ljunggren et al., 2016).

2.2.3 Sistemas comerciais e patentes de dispositivos de medição de sinais vitais

Os sinais vitais constituem fortes indicadores da condição clínica de um doente, e permitem detetar situações de crise antecipadamente, pelo que a sua monitorização permite responder a alarmes e, assim, prevenir situações de deterioração, ataques cardíacos e morte. Neste sentido, o uso de equipamentos eletrónicos capazes de monitorizar sinais vitais constituem uma grande

melhoria no processo de recolha e subsequentemente permitem reduzir a mortalidade (Griffiths, Saucedo, Schmidt, & Smith, 2015).

Relativamente a este tipo de equipamentos eletrônicos, importa referir que a tendência é que sejam cada vez mais pequenos e possíveis de integrar em peças de vestuário para que o utilizador possa monitorizar a sua atividade durante longos períodos de tempo. Alguns destes dispositivos permitem ainda, não só o uso em ambiente clínico para diagnóstico médico, mas também o uso diário em ambiente doméstico e aplicações de *fitness*, e possibilitam monitorizar sinais vitais, bem como outros parâmetros relevantes (Khan et al., 2016) (Bonato, 2005). A Tabela 2 resume as principais funcionalidades de alguns sistemas comerciais e patentes de monitorização de sinais vitais com diversas aplicações.

Tabela 2 – Sistemas comerciais e patentes de medição de sinais vitais com diversas aplicações.

Solução de monitorização	Fabricante	Medição de parâmetros							Alertas	Localização	
		FC	TC	SpO ₂	TR	PA	P	G		Indoor	GPS
<i>ViSi Mobile System</i>	<i>Sotera Wireless, Inc.</i>	✓	✓	✓	✓	✓	✓		✓		
<i>Aingeal</i>	<i>Intelesens</i>	✓	✓		✓				✓		
<i>EQ02</i>	<i>Equivital</i>	✓	✓	✓	✓	✓	✓	✓			✓
<i>VitalMote</i>	-	✓		✓		✓				✓	✓
Patente US 20140085082 A1	<i>Physio-Control, Inc.</i>	✓		✓	✓	✓			✓		
Patente US 8956293 B2	<i>Sotera Wireless, Inc.</i>		✓	✓	✓	✓			✓	✓	

✓ – Apresenta esta funcionalidade

FC – Frequência Cardíaca

TC – Temperatura Corporal

SpO₂ – Saturação do Oxigénio no Sangue

TR – Taxa Respiratória

PA – Pressão Arterial

P – Pulsação

G – Resposta Galvânica da Pele

De uma forma geral, descrevem-se assim as principais características dos sistemas apresentados, bem como o objetivo da sua utilização.

- ***ViSi Mobile System***: sistema desenvolvido para monitorização de doentes em ambiente hospitalar, permitindo avaliar os quatro sinais vitais e ainda a pulsação e a pressão arterial, e alertar os profissionais de saúde caso ocorra alguma alteração. O objetivo é a diminuição das medições exaustivas de sinais vitais pelos profissionais de saúde, e a constante monitorização para detetar alterações entre cada medição. É um dispositivo pequeno usado no pulso, e sem fios, embora algumas medições exijam o uso de sensores adicionais, como eletrodos (“*ViSi Mobile System*”).
- ***Aingeal***: sistema capaz de medir as frequências cardíaca e respiratória e a temperatura corporal, e ainda traçar um eletrocardiograma, enviando estas informações através de *Wi-Fi* para uma estação de vigilância capaz de registar

alarmes e de os listar por ordem de prioridade (“ECG Wireless Hospital Monitoring Solution,” 2016).

- **EQ02:** é usado no peito com a ajuda de um cinto, e para além das funcionalidades dos dois dispositivos anteriores, mede ainda a resposta galvânica da pele, tem *Global Positioning System* (GPS) e deteção de quedas. Para além de um *software* de visualização dos dados, esta solução tem ainda uma aplicação para *smartphone*, com o qual comunica através de *Bluetooth* (“Sense and Transmit,” 2017).
- **VitalMote:** dispositivo não só de monitorização, mas também de localização *indoor* e *outdoor*, com o objetivo de monitorizar e aceder à localização de doentes após a triagem feita por paramédicos até à chegada ao serviço de urgência (Gao, Greenspan, Welsh, Juang, & Alm, 2005).
- **Patente US 20140085082 A1:** monitorização de um doente através de uma pulseira usada pelo profissional de saúde, que pode aceder aos parâmetros fisiológicos do doente (Lyon, Walker, Edwards, Andersson, & Johnson, 2013).
- **Patente US 8956293 B2:** sistema constituído por sensores individualizados de monitorização de sinais vitais, que têm em conta na medição o estado do doente (movimento, descanso ou queda). Este sistema foi desenvolvido para ambiente hospitalar, e tem ainda um *software* capaz de gerar alarmes e um sensor de localização via *wireless* que dispõe de um mapa na interface do utilizador (McCombie, Dhillon, Banet, Trommer, & Moon, 2009).

Após esta análise, resta estudar alguns métodos de localização *indoor* e quais as tecnologias em que estes se podem aplicar, de forma a ser possível a localização de um doente num serviço de urgência.

2.3 Localização *indoor*

Os sistemas de localização *indoor*, ou *Indoor Positioning Systems* (IPS), são sistemas que permitem localizar pessoas ou objetos em ambiente *indoor*, ou seja, dentro de edifícios. As aplicações mais recentes incluem o posicionamento em centros comerciais, permitindo localizar lojas e analisar o fluxo de pessoas através do *smartphone*, e em algumas delas encontrar os produtos desejados; aeroportos, dando acesso ao utilizador ao caminho mais curto para chegar a determinada porta de embarque e qual o tempo estimado do percurso; museus, não só para indicar percursos mas incluindo também áudio, vídeos e informações acerca do ponto onde se encontra o utilizador; e em hospitais, ajudando o utilizador a orientar-se dentro do hospital, determinando qual a localização de profissionais de saúde ou doentes e localizar equipamentos e materiais em tempo real.

Os IPS funcionam de forma semelhante ao GPS; no entanto, o GPS não pode ser usado *indoor* por sofrer atenuação do sinal de satélite devido aos telhados e paredes dos edifícios, causando uma perda de precisão (Senion, 2016).

A localização *indoor* de pessoas ou objetos pode ser conseguida através de diferentes métodos alternativos ao GPS e diversas tecnologias, com diferentes níveis de precisão e de forma mais ou menos simples. Algumas destas tecnologias são atualmente usadas diariamente para outros fins ou até noutras aplicações que não requerem um sistema de localização; no entanto, podem facilmente ser adaptadas a este tipo de sistemas.

Ao longo desta secção serão apresentados alguns métodos que podem ser aplicados na localização *indoor* (Secção 2.3.1), e ainda algumas tecnologias que podem ser usadas (Secção 2.3.2), para que seja possível efetuar uma comparação entre métodos e tecnologias (Secção 2.3.3). Serão ainda dados alguns exemplos de sistemas comerciais ou de patentes que permitam a localização de pessoas no interior de edifícios (Secção 2.3.4).

2.3.1 Métodos de localização *indoor*

Existem essencialmente dois métodos de localização *indoor*: o método de *fingerprinting* (Secção 2.3.1.1) e o método de trilateração (Secção 2.3.1.2), bastante diferentes entre si, mas que no entanto podem recorrer ao mesmo tipo de equipamentos e sinais. Podem ainda ser utilizados por diferentes tecnologias, ainda que a mesma tecnologia possa permitir a implementação dos dois métodos. O essencial na decisão por um destes métodos inclui a sua precisão, ou em casos em que esta não é o ponto mais relevante, a sua facilidade de implementação e utilização, bem como os custos que esta exige.

2.3.1.1 *Fingerprinting*

A técnica de localização por *fingerprinting* é uma técnica que permite aproveitar redes locais já existentes em edifícios para a localização *indoor*, pelo que constitui uma alternativa segura à utilização de GPS (K.Kaemarungsi & P.Krishnamurthy, 2004).

O método de localização por *fingerprinting* é constituído por duas fases distintas. Numa primeira fase é necessário colocar em locais estratégicos e conhecidos dispositivos com sensores capazes de efetuar a recolha de *fingerprints*, cujos valores obtidos serão inseridos numa base de dados (Kárník & Streit, 2016). Pode, assim, obter-se uma visualização da área em que o sistema é implementado dividida numa grelha de pontos, na qual se podem marcar os pontos recolhidos durante o treino do sistema, já inseridos na base de dados. Numa segunda fase, uma fase *online*, pode proceder-se à recolha do *Received Signal Strength Indicator* (RSSI) de pontos móveis por vários pontos de acesso, que serão posteriormente enviados para um servidor (K.Kaemarungsi & P.Krishnamurthy, 2004). O servidor é capaz de estimar a localização aproximada destes pontos por comparação com os vetores de *fingerprinting* guardados na base de dados, devolvendo ao utilizador o valor mais aproximado (Kárník & Streit, 2016).

Esta técnica, de acordo com Kornuta (Kornuta, Acosta, & Toloza, 2013), é a técnica mais utilizada por apresentar um nível elevado de precisão, tendo um erro de apenas 2-3 metros em ambiente *indoor*, bem como baixos custos de implementação.

2.3.1.2 Trilateração

Em oposição ao método anterior, o método de trilateração consiste num modelo matemático em que é calculada a localização de um ponto no espaço a partir da distância a que este ponto se encontra relativamente a um transmissor. A partir da distância entre o transmissor o ponto de interesse que se pretende localizar é possível traçar uma circunferência com centro no transmissor e raio igual à distância, podendo o ponto de interesse estar localizado em qualquer local desta circunferência.

Para obter uma localização mais precisa, podem ser usados dois transmissores (trilateração 2-D); no entanto, neste caso, iria existir ambiguidade proveniente do facto de que o círculo de cada um dos transmissores se pode intersectar, havendo assim duas localizações possíveis nas quais o ponto de interesse pode estar. A solução será então usar não dois mas três transmissores (trilateração 3-D), sendo que neste caso o ponto de interesse se encontra na área de intersecção entre os círculos destes (Circuit Cellar Staff, 2014). A Figura 5 é uma representação dos métodos de trilateração descritos. É de ter em atenção que os círculos identificados com A, B e C representam transmissores, e os identificados com P, P1 e P2 são pontos de interesse.

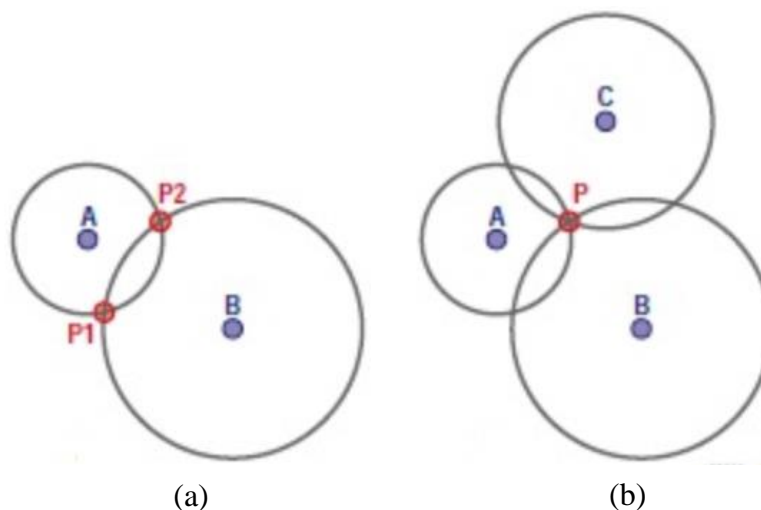


Figura 5 – Técnicas de trilateração. (a) Trilateração 2-D. (b) Trilateração 3-D (Fonte: (Circuit Cellar Staff, 2014)).

A distância a que o ponto de interesse se encontra em relação ao transmissor pode ser calculada de várias formas, das quais se destaca a medição da força do sinal recebido (RSSI), usado também na técnica de *fingerprinting*, o tempo de chegada (*ToA*), dado pelo tempo necessário a que um sinal de radiofrequência enviado por um transmissor seja recebido pelo recetor, e o ângulo de chegada (*AoA*), baseado na direção segundo a qual o sinal é recebido (Kárník & Streit, 2016).

Os dois métodos descritos (*fingerprinting* e trilateração) são atualmente aplicados por tecnologias utilizadas em sistemas de localização *indoor*, fazendo-se assim uma abordagem a diferentes tecnologias na Secção 2.3.2.

2.3.2 Tecnologias usadas em sistemas de localização *indoor*

Várias tecnologias podem ser usadas em IPS, e a escolha reside não só na acessibilidade e no preço, mas também em fatores como o alcance, a precisão e a segurança. As tecnologias mais usadas incluem *Wi-Fi*, *Bluetooth Low Energy* (BLE) e *Radio Frequency Identification* (RFID) (“*Indoor location technologies compared*,” 2015). Uma outra tecnologia que pode ser utilizada é a tecnologia *Zigbee*, que tal como o *Bluetooth*, é uma tecnologia de curto alcance e baixo consumo de energia (Kárník & Streit, 2016). De seguida estas principais tecnologias serão abordadas e descritas, bem como outras tecnologias menos utilizadas em localização *indoor*.

2.3.2.1 *Wi-Fi*

Atualmente todos os computadores têm capacidade de estabelecer comunicações sem fios através de *Wi-Fi*, que possibilita a que as pessoas possam aceder à rede de forma fácil, estando por isso “espalhada” por todo o lado: hotéis, cafés, aeroportos, entre outros (Poole). Na realidade estes sinais são ondas eletromagnéticas formadas quando a energia elétrica atravessa uma peça metálica, como um fio ou uma antena, e podem viajar por determinadas distâncias, dependendo da força do sinal. Estas ondas são capazes de transmitir informações em formato de áudio, vídeo, voz e dados (Gast, 2005).

A tecnologia *Wi-Fi* é regulada pela norma IEEE802.11, que visa vários protocolos, que diferem principalmente pela banda de frequência em que operam e pela capacidade de transmissão de dados (Poole). Esta norma foi criada em 1997, devido à necessidade de regulamentar a utilização do *Wi-Fi* por utilizadores individuais, autorizada em 1985. Atualmente é regulada pela *Wi-Fi Alliance*, uma organização global sem fins lucrativos (Britannica, 2016). A Tabela 3 apresenta as principais características destes protocolos.

Tabela 3 - Protocolos da norma IEEE 802.11 (Fonte: (Poole) (Harnwood, 2009)).

	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g	IEEE 802.11n
Velocidade máxima de transmissão de dados (Mbps)	54	11	54	600
Velocidade típica de transmissão de dados (Mbps)	25	5	-	-
Alcance médio <i>indoor</i> (m)	~30	~30	~46	~53
Banda RF (GHz)	5	2.4	2.4	2.4 ou 5
Largura do canal (MHz)	20	20	20	20 ou 40

A tecnologia *Wi-Fi*, como método de localização *indoor*, constitui uma grande vantagem, uma vez que na grande maioria dos edifícios e infraestruturas já existem pontos de acesso, que podem ser usados, não havendo assim a necessidade de *hardware* adicional.

Através do cálculo da força do sinal (ou RSSI) é possível estabelecer um método de *fingerprinting*, sendo apenas necessária a instalação de uma aplicação no *smartphone* que calcule a posição do utilizador atendendo aos dados recolhidos (Gaudlitz, 2015).

Dado que a norma IEEE 802.11n é a que apresenta a maior velocidade de transmissão, o maior alcance e a maior largura de banda é a norma que vai ser considerada para uma comparação entre tecnologias de localização. Importa ainda referir que existem determinadas frequências do espectro de radiofrequências que são reservadas a aplicações industriais, científicas e médicas (banda *Industrial, Scientific and Medical (ISM)*). Em Portugal, a banda reservada à comunicação sem fios, incluindo sistemas de monitorização, é a banda dos 2.4GHz, sendo assim a tecnologia com frequência mais indicada para aplicações relativas a cuidados de saúde (Clemente, 2012) (Vallejos de Schatz, Medeiros, Schneider, & Abatti, 2012).

2.3.2.2 Zigbee

Zigbee é um conceito associado a uma tecnologia de rede *wireless* com um padrão global aberto que permite uma comunicação de baixo custo, baixo consumo e de curto alcance. É regulado pela *Zigbee Alliance* segundo a norma IEEE 802.15.4, criada em 2003 (“ZigBee® Wireless Standard,” n.d.).

Uma rede *Zigbee*, na sua forma mais complexa, constitui uma topologia *mesh*, ou seja, uma rede emalhada. Nesta topologia cada nó comunica com outro nó através de ondas rádio, existindo três tipos de nós: o *Zigbee Coordinator*, único, responsável pela criação da rede, e que permite que outros nós se liguem a ele, podendo enviar e receber mensagens; os *Zigbee Router*, ligados ao anterior, que permitem alargar a rede, podendo interligar dispositivos fora do alcance de comunicação entre si e recalculam percursos de comunicação caso haja uma falha em algum dos caminhos existentes; os *Zigbee End Device*, capazes de receber e enviar mensagens através dos dois dispositivos anteriores, e podendo ser desligados sem que nada na rede sofra alterações. Isto faz com que uma rede *Zigbee* seja flexível, alargada e eficiente em termos de comunicação entre nós e dispositivos.

Este tipo de tecnologia possibilita um grande leque de aplicações, desde a substituição de cabos ponto a ponto a sistemas de segurança, casas inteligentes em que se podem controlar luzes, portas e estores, entre outros, automação industrial e monitorização hospitalar (NXP Laboratories UK, 2012). Nesta última aplicação inclui-se a monitorização de doentes com doenças crónicas, a monitorização periódica de alguns indicadores de saúde, como a frequência cardíaca e a temperatura corporal, e a combinação da monitorização dos sinais vitais do doente com a monitorização do ambiente em que este se encontra, com detetores de movimento e de controlo ambiental, entre outros (Alliance, 2009).

Relativamente a esta tecnologia, importa ainda referir que este protocolo pode operar em diferentes bandas de frequência, permitindo um diferente número de canais e diferentes velocidades. Estes dados encontram-se na Tabela 4.

Tabela 4 - Características de funcionamento da tecnologia *Zigbee*.

Banda de Frequência	Canais de Operação	Velocidade (kbps)
2.4 GHz	16 (canais 11-26)	250
915 MHz	10 (canais 1-10)	40
868 MHz	1 (canal 0)	20

Um sistema de localização *indoor* baseado em *Zigbee* pode ser implementado de forma simples, recorrendo aos nós estáticos da rede, ou seja, os nós de localização conhecida e plantados em locais estratégicos. Usando uma técnica de trilateração ou multilateração (no caso de existirem mais do que três nós estáticos) é possível identificar um nó em movimento através do seu RSSI (Hernandez & Jain, 2009).

Mais uma vez, a banda de frequência a considerar serão os 2.4GHz, para a qual, de acordo com a Tabela 4, a tecnologia *Zigbee* apresenta 16 canais disponíveis e uma velocidade de transmissão de dados de 250 kbps.

2.3.2.3 *Bluetooth* e BLE

O *Bluetooth*, tal como as tecnologias anteriormente descritas, é uma tecnologia de comunicação sem fios e de curto alcance, a operar na banda dos 2.4 GHz (“How does *Bluetooth* work?”, 2007). Este tipo de tecnologia utiliza ondas de rádio para comunicar, partindo do pressuposto de que os dispositivos em questão se encontram emparelhados (“What is *Bluetooth*?”). A topologia utilizada é a topologia em estrela, também chamada “*Piconet*”, na qual existe apenas um dispositivo mestre e podem existir até sete dispositivos escravos. Cada dispositivo pode fazer parte de várias redes com esta configuração, podendo ser simultaneamente mestre e escravo. Na comunicação é usada uma frequência específica, determinada pelo dispositivo mestre, sendo possível o uso de 79 frequências diferentes da banda ISM (Foley, 2007).

A tecnologia *Bluetooth* surgiu em 1994 apenas com o propósito de eliminar os fios na comunicação entre dispositivos; o ano 2000 foi o ano em que os primeiros dispositivos com *Bluetooth* começaram a ser comercializados. Nos últimos anos verifica-se que todos os *smartphones* têm capacidade de comunicação *Bluetooth*, pelo que se tornou no tipo de comunicação padrão para a conexão com objetos como colunas, *headphones* e outros acessórios sem fios (Nield, 2016). Atualmente é ainda possibilitada a conceção de sensores cada vez mais pequenos e capazes de funcionar durante períodos de tempo mais longos através da tecnologia *Bluetooth Low Energy* (BLE) (“What is *Bluetooth*?”). Para além de um consumo de energia bastante inferior relativamente ao *Bluetooth* clássico, o BLE proporciona tempos de conexão de apenas alguns milissegundos, uma velocidade de transmissão de dados bastante superior, a possibilidade de um maior número de dispositivos escravos ligados na mesma rede e um *Attribute Protocol* (ATT), que permite que se possam aceder às características do dispositivo e usá-las nas mais variadas aplicações (“*Bluetooth Low Energy Technology*”). Estas características fazem com que atualmente o BLE possa ser usado em

aplicações como casas inteligentes, sistemas de segurança, aplicações de bem-estar e *fitness* e sensores de diversos tipos (“What is *Bluetooth*?”).

Os sistemas de localização *indoor* baseados na tecnologia *Bluetooth* são possíveis através da utilização de *beacons*, pequenos transmissores rádio de baixo custo e facilmente instaláveis (“Indoor Positioning, Tracking and Indoor Navigation with Beacons”), cujo sinal emitido pode ser detetado pelos dispositivos móveis, que calculam a distância à sua fonte, dando assim a localização aproximada em que o utilizador se encontra (Blaz, 2015).

2.3.2.4 RFID

A tecnologia RFID é uma forma de comunicação sem fios que, tal como as tecnologias analisadas anteriormente, usa ondas de rádio para comunicar. Pressupõe a utilização de dois componentes distintos: uma *tag*, que se encontra unida ao elemento que se pretende identificar, e um leitor, fixo ou móvel, conectado à rede, e que envia potência à *tag* através de uma pequena antena com o objetivo de efetuar a leitura dos dados da *tag* ou enviar comandos (“RFID Technology Primer”).

Esta tecnologia surgiu em 1970 como forma de substituir a tecnologia de código de barras, que apresentava algumas desvantagens de utilização em gestão e manutenção de produtos desde o processo de fabrico. A tecnologia RFID surgiu como uma grande melhoria à tecnologia já existente na medida em que, ao contrário dos códigos de barras, permite não só a leitura mas também a escrita de dados, pelo que os dados de uma *tag* podem ser alterados, atualizados e bloqueados (Bonsor & Fenlon, 2007). As suas aplicações atualmente incluem o inventário automatizado e o controlo de *stock*, a localização de pessoas e objetos na área dos cuidados de saúde e da indústria, o bloqueio/desbloqueio remoto de equipamentos eletrónicos e aplicações de segurança (“RFID Technology Primer”).

A localização *indoor* com recurso à tecnologia RFID pode ser repartida em duas vertentes: a localização de uma *tag* alvo utilizando três ou mais leitores, dada por trilateração, ou utilizando leitores com diferentes níveis de potência emitida e colocados em locais específicos, que recorrem a técnicas e filtros próprios; a colocação de *tags* em locais definidos, que quando detetadas pelo leitor em movimento, permitem estimar a sua localização a partir de métodos matemáticos. A grande desvantagem na utilização desta tecnologia na localização *indoor* é a sua suscetibilidade a interferências, pelo que é necessário um estudo intensivo quando se pretende implementar sistemas com recurso a RFID (Papapostolou & Chaouchi, 2011).

2.3.2.5 Outras tecnologias

Para além das tecnologias enunciadas, que são as mais atuais e mais utilizadas, é possível a localização *indoor* através de outras tecnologias menos usuais, como *Ultra-Wide Band*

(UWB), *Pedestrian Dead Reckoning* (PDR), som e ultrassom e técnicas de análise de imagem.

A tecnologia UWB é uma tecnologia sem fios capaz de transmitir informação através de uma banda de radiofrequências larga e a uma potência capaz de ultrapassar obstáculos como portas ou paredes (Rouse, 2008). Quando operando na banda dos 5GHz pode atingir velocidades de transmissão de cerca de 1.6 Gbps, bastante superior à velocidade máxima das tecnologias Wi-Fi e Zigbee, como comprovam a Tabela 3 e a Tabela 4. No entanto, com o aumento da distância, esta velocidade decai drasticamente. Pode ser comparável com a tecnologia *Bluetooth* e seria facilmente integrada em dispositivos deste tipo, tendo no entanto sido substituída por este e por alguns protocolos da tecnologia *Wi-Fi* (Mitchell, 2016). Calculando o tempo de chegada do sinal emitido pelo transmissor é possível obter a distância entre um ponto de referência e o alvo a localizar (Alarifi et al., 2016).

Uma outra tecnologia suscetível de ser utilizada na localização *indoor* é a tecnologia PDR. Esta tecnologia recorre ao uso de sensores inerciais como acelerómetros, giroscópios e compassos digitais (Lan & Shih, 2014) com o intuito de determinar a localização do utilizador através da sua última posição e da velocidade à qual este se desloca, podendo ser definida sob a forma de coordenadas cartesianas (Alarifi et al., 2016). Uma vantagem na utilização desta tecnologia é que, atualmente, os sensores necessários já se encontram presentes na maioria dos *smartphones*, pelo que a sua utilização se torna mais simples. Não requer ainda o uso de sensores adicionais, ao contrário da tecnologia *Bluetooth*, por exemplo, que exige a utilização de *beacons* (DeMoes, 2016).

A localização *indoor* pode ainda ser conseguida através da utilização de som ou ultrassom. Esta técnica recorre à velocidade do som, o que a torna imprecisa na medida em que a velocidade de propagação do som no ar é influenciada pela humidade e pela temperatura, e ainda pelo ruído, embora a sua precisão seja da ordem do centímetro e possa detetar múltiplos objetos em movimento de forma simultânea (Ijaz & Yang, 2013).

Por fim, existem os métodos de análise de imagem, que essencialmente são sistemas de visão por computador que permitem a localização de pessoas através da análise de imagens recolhidas por câmaras. Embora permitam a localização de várias pessoas simultaneamente e previnam a perda de identidade ao serem usadas imagens a cores, é necessário expor os pontos mortos e as oclusões, o que o tornam um método bastante caro devido à necessidade de *hardware* (Koyuncu & Yang, 2010).

Como referido anteriormente, as tecnologias abordadas neste ponto não são tão usuais ou apresentam uma maior dificuldade na implementação, pelo que na comparação que será apresentada apenas serão consideradas *Wi-Fi*, *Zigbee*, *Bluetooth* e RFID.

2.3.3 Comparação entre métodos e tecnologias usados em sistemas de localização *indoor*

Após uma pequena introdução aos métodos e a várias tecnologias que podem ser usados em sistemas de localização *indoor*, importa construir uma análise comparativa entre eles para que

se possa inferir qual será o método e a tecnologia que apresenta mais vantagens de utilização, nomeadamente em ambiente hospitalar, por ser o ambiente no qual deverá ser aplicado o sistema a desenvolver.

De acordo com um estudo que compara diferentes métodos de localização *indoor* em ambiente hospitalar (Van Haute et al., 2016), o método que apresenta uma maior precisão é o método de *fingerprinting*, em que o erro máximo foi de 1,99 metros. A comparação foi realizada entre o método de *fingerprinting* e dois outros métodos, sendo que um deles recorria ao *ToA* e o segundo recorria ao RSSI e à distância entre o emissor e o recetor. No entanto, este método requer uma fase de calibração exaustiva e que, perante alterações significativas no ambiente no qual este método é usado, poderá ter de ser repetida, o que se torna num processo demorado (Van Haute et al., 2016).

Relativamente à tecnologia mais favorável a ser usada, foi realizada uma tabela comparativa (Tabela 5) em que os fatores a ter em conta, numa primeira análise, são a banda de radiofrequência em que operam, o alcance em ambiente *indoor*, a velocidade de transmissão de dados e o consumo energético.

Tabela 5 – Estudo comparativo entre as tecnologias mais utilizadas em sistemas de localização *indoor* (Fonte: (Poole)).

	Banda RF (GHz)	Alcance <i>indoor</i> (m)	Velocidade de transmissão de dados (Mbps)	Consumo de Energia
Wi-Fi	2.4 ou 5	~53	600	Elevado
Zigbee	2.4	Até 70	0.25	Baixo
Bluetooth Low Energy	2.4	Até 10	1	Baixo
RFID	Várias (2.4GHz em SHF)	Superior a 3 para SHF 10 para UHF	0.64 para UHF	Baixo

É de notar que, para qualquer tecnologia para a qual seja possível calcular o RSSI, se pode usar o método de *fingerprinting*, podendo neste caso ser usado com a tecnologia *Wi-Fi*, *Bluetooth* e *Zigbee*. Recorrendo ainda ao estudo já mencionado, a tecnologia que apresentou melhores resultados, independentemente do método usado, foi *Wi-Fi*, embora as restantes tecnologias mostrem resultados semelhantes (Van Haute et al., 2016). Um artigo publicado pela *connectBlue* (Svensson & Saltzstein, n.d.) comprova que a tecnologia *Wi-Fi* é a mais usualmente escolhida quando se pretende inserir um sistema num hospital usando a rede de comunicações já existente. No entanto, quando se constrói uma rede de raiz, o *Bluetooth* é uma boa opção, tendo no entanto uma limitação no alcance, comparativamente com a tecnologia *Zigbee*. Por sua vez, esta tem uma velocidade de transmissão de dados baixa relativamente às anteriores; no entanto, de acordo com Vallejos de Schatz (Vallejos de Schatz et al., 2012), esta velocidade é suficiente para monitorizar sensores que medem parâmetros médicos. Ainda segundo esta referência, o melhor sistema seria usar a tecnologia *Zigbee* para

conectar com dispositivos próximos a um baixo custo, e usar o *Wi-Fi* para comunicações de longa distância e com maior requisito relativamente à largura de banda para armazenar dados.

2.3.4 Sistemas comerciais e patentes de sistemas de localização *indoor*

Atualmente existem já vários sistemas de localização *indoor*, alguns dos quais permitem ao utilizador a adaptação ao ambiente no qual se encontra inserido através de portais *Web* e a visualização de dados através do *smartphone*. Recorrem ainda aos diferentes métodos abordados, bem como a diferentes tecnologias.

Desenvolvido pelo MIT em 2016, existe um sistema de localização *indoor* que utiliza a tecnologia *Wi-Fi*. Este sistema, denominado *Chronos*, utiliza um único ponto de acesso para localizar utilizadores com precisão na ordem do decímetro. O método usado na medição da distância entre o transmissor e o recetor, ambos dispositivos *Wi-Fi*, é o tempo de voo (*ToF*) (Vasisht, Kumar, & Katabi, 2016).

Recorrendo à tecnologia *Wi-Fi*, existe um sistema desenvolvido pelo Laboratório de Automática e Sistemas do Instituto Pedro Nunes. Este sistema foi desenvolvido com o propósito de ajudar pessoas idosas, permitindo aos profissionais que lhes prestam assistência conhecer a sua posição para, assim, os poder auxiliar em caso de necessidade. Este sistema utiliza pontos de acesso *Wi-Fi*, cujos sinais emitidos podem ser lidos por um telemóvel que, a partir da rede *Wi-Fi*, comunica com um servidor, que lhe devolve a sua localização baseada na força dos sinais lidos (Quintas, Cunha, Serra, Pereira, & Marques, 2013). Este foi o sistema adotado neste projeto, tal como explicado na Secção 4.3.2.

Usando a tecnologia *Zigbee* existem várias patentes, das quais um sistema de localização *indoor* em tempo real que utiliza um conjunto de *tags* colocadas nos objetos ou pessoas que se pretende localizar. O cálculo da localização é feito através de um computador, que recolhe os dados obtidos dos *routers Zigbee* e do nó mestre, permitindo ainda a visualização dos dados (Garcia, Nunez, Saavedra, & Tapia, 2010).

Recorrendo à tecnologia *Bluetooth* existe a *Estimote*, que usa vários dispositivos de *hardware iBeacon* dispostos no centro das paredes de um determinado local. A planta do local é desenhada a partir do *smartphone* do utilizador, tendo este de se deslocar entre os pontos o mais próximo possível das paredes. A marca disponibiliza ainda um *Software Development Kit* (SDK) de desenvolvimento para *iOs*, permitindo ao utilizador construir o seu próprio *software* (Estimote, n.d.). Outro exemplo é a *proximi.io* e o seu sistema de localização por *Bluetooth* usando um método de trilateração. A sua plataforma também permite o desenvolvimento por parte do utilizador (Proximi.io, n.d.).

Um outro sistema comercial é o *IndoorAtlas*, que usa um método de localização *indoor* baseado em tecnologia geomagnética, ou seja, é usado o campo magnético único de um edifício e os sensores comuns de um *smartphone* para localização *indoor*. Não é requerida nenhuma compra de *software* e não exige manutenção em larga escala. A precisão de localização é de 1-2 metros, e é possível aliar a tecnologia *Wi-Fi* (ou outras) para otimização.

Esta plataforma é ainda compatível com *iOS* e *Android* e é capaz de dar a localização de um dispositivo em tempo real (*IndoorAtlas*, n.d.).

Para além destes exemplos existem vários outros, que não foram abordados. Em suma, a grande maioria dos sistemas comerciais recorre à tecnologia geomagnética, pois destina-se ao uso em *smartphones*, e na maior parte dos casos alia-se às tecnologias *Wi-Fi* e *Bluetooth* para uma maior interação por parte do utilizador e para a obtenção de melhores resultados.

2.4 Considerações finais

O elevado afluxo de doentes nas urgências dos hospitais portugueses e a presença de um grande número de situações pouco urgentes ou não urgentes pode representar um problema, uma vez que, principalmente em épocas típicas de gripes, pode conduzir a tempos de espera demasiado longos de acordo com a triagem de Manchester. Entre o momento da triagem e a primeira observação do médico a condição de saúde do doente pode ser alterada e, se não for detetada atempadamente essa alteração pelo profissional de saúde, o doente pode perecer ou sofrer complicações evitáveis. Apesar de terem sido tomadas algumas medidas para evitar os picos de procura dos serviços de urgência portugueses, estes continuam a ser procurados por grande parte da população, não tendo estas medidas sido tão eficazes como seria desejável.

A solução apresentada passa pelo desenvolvimento de um sistema de monitorização de sinais vitais, que permita que o doente seja monitorizado em tempo real durante o episódio de urgência, e que sejam lançados alertas ao profissional de saúde que possam prever uma situação de crise. A integração do algoritmo de localização permite ao profissional de saúde saber exatamente e de forma rápida a localização do doente em determinado momento, minimizando assim o tempo de assistência em caso de necessidade.

Analisando a importância dos sinais vitais (Secção 2.2.2) pode verificar-se que todos eles são relevantes para a deteção precoce de alterações do estado de saúde. No entanto, e atendendo aos métodos de medição apresentados para cada um destes sinais, nem todos podem ser monitorizados através de métodos simples e recorrendo a dispositivos pequenos e facilmente convertidos em *wearables*. Neste caso, os sinais vitais que podem facilmente ser adquiridos são a frequência cardíaca, sendo que existem diversos dispositivos pequenos a usar a fotopleitismografia, como é o caso das pulseiras de *fitness*, e a temperatura corporal, medida através de um simples sensor de temperatura. Uma vez que não se pretende usar acessórios e sensores adicionais considera-se que a frequência cardíaca e a temperatura corporal são os únicos parâmetros vitais a considerar no desenvolvimento do protótipo. Refere-se ainda que a medição destes sinais vitais foi sugerida pela empresa *Glintt* e adotada neste projeto, tal como outras sugestões apresentadas no Capítulo 3. É ainda de salientar que, de acordo com a informação da Tabela 2, nenhum dos sistemas comerciais ou patentes encontradas reúnem a monitorização destes dois parâmetros, um algoritmo de localização *indoor* e a capacidade de emitir alertas. Também nenhum destes sistemas foi projetado para uso em ambiente de urgência hospitalar, o que reforça que o protótipo desenvolvido constitui um sistema inovador nesta área.

A localização *indoor* por *fingerprinting* foi considerado o melhor método a usar, e a tecnologia *Wi-Fi* a melhor tecnologia, pelo que se optou pelo uso conjunto destes dois.

3. Requisitos do sistema e arquitetura

O sistema a desenvolver visa a monitorização de sinais vitais de um determinado doente em ambiente hospitalar de urgência, sendo estes sinais vitais recolhidos através de uma aplicação *Android* instalada num *smartphone* e guardados numa base de dados alojada num servidor, para que possam ser usados pelo *software* a implementar. Por sua vez, este *software* deverá permitir toda a gestão dos dados de cada doente individualmente, incluindo a visualização dos dados recolhidos.

Na Secção 3.1 são identificados os requisitos do sistema de forma a corresponder às necessidades do utilizador. De seguida, descrevem-se as *user stories* do sistema (Secção 3.2) sob o ponto de vista do doente como utilizador final (Secção 3.2.1) e do profissional de saúde (Secção 3.2.2). As especificações técnicas para o *software* são analisadas na Secção 3.3, onde se apresentam alguns *mockups* desenhados com o objetivo de dar uma visão geral do sistema após a sua implementação. Na Secção 3.4 analisam-se as linguagens de programação possíveis de utilizar no desenvolvimento do sistema proposto. De seguida, na Secção 3.5, é apresentada a arquitetura do *software*, sendo analisados o diagrama de componentes (Secção 3.5.1), o diagrama de atividades (Secção 3.5.2) e alguns diagramas de sequência (Secção 3.5.3) do sistema. Na Secção 3.6 analisam-se as especificações técnicas do *hardware* a utilizar para medição dos sinais vitais do doente. Por fim, a Secção 3.7 apresenta algumas considerações finais acerca deste capítulo, abordando os *softwares* e metodologias utilizadas.

3.1 Identificação dos requisitos

O sistema a desenvolver terá de cumprir determinados requisitos de forma a providenciar uma boa experiência, tanto para o doente como para o profissional de saúde, que são os utilizadores finais da solução. Torna-se, assim, importante definir um conjunto de exigências a que cada componente do sistema terá de obedecer, atendendo às necessidades do utilizador final. Esta análise inclui qual terá de ser o seu comportamento, quais as funcionalidades que terá de ter de acordo com o que o utilizador pretende fazer e qual a arquitetura desejada e adequada para cumprir os requisitos impostos.

De uma forma sucinta, os requisitos do sistema são os seguintes:

- O dispositivo de monitorização deverá ser capaz de medir a frequência cardíaca do doente, bem como a sua temperatura corporal, em tempo real;
- O dispositivo de monitorização deverá ser localizável em tempo real;
- O dispositivo de monitorização deverá comunicar com um dispositivo móvel, transmitindo os dados recolhidos;
- O dispositivo móvel deverá comunicar com um servidor via *wireless*;
- O servidor deverá guardar os dados recebidos;
- O servidor deverá ser capaz de fornecer dados à *dashboard* do sistema;
- A *dashboard* deverá ser simples de utilizar;

- A *dashboard* deverá permitir inserir os dados de um novo doente no sistema ou excluir estes dados quando necessário;
- A *dashboard* deverá permitir visualizar os dados de cada doente individualmente, incluindo alguns dados pessoais, os sinais vitais que são recolhidos e a sua localização em tempo real, e gerar alertas quando os sinais vitais do doente apresentarem alterações significativas.

É importante referir que estes requisitos pretendem ir de encontro aos objetivos do projeto e às funcionalidades pretendidas.

Após esta análise de requisitos relativamente às características de cada um dos componentes do sistema, torna-se relevante saber que funcionalidades são desejáveis por parte do utilizador final, e que capacidades o sistema deverá permitir. Parte-se, assim para uma análise de *user stories*.

3.2 *User stories*

O desenvolvimento de *software* através da metodologia *Agile*, segundo a Optimus Information (Optimus Information, 2016), é um novo método de desenvolvimento que pretende ser mais flexível, mais barato e com uma maior interação por parte do utilizador do que os tradicionais métodos de desenvolvimento, uma vez que este participa em todas as fases da conceção (Optimus Information, 2016). O objetivo desta metodologia é desenvolver um projeto de forma gradual, dividindo-o em pequenas tarefas e funcionalidades que se vão incrementando desde o início do desenvolvimento até ao produto final. No início do projeto é feita uma lista de funcionalidades que o *software* deverá ter atendendo ao que é pretendido pelo cliente. A estas funcionalidades é dado o nome de *user stories*, e definem as tarefas a realizar durante o desenvolvimento. De seguida deverá ser atribuída uma linha temporal para cada *user story* e definidas prioridades. Importa referir que ao longo do tempo a lista é variável, atendendo ao tempo de desenvolvimento (Rasmusson, n.d.).

As *user stories* são, assim, pequenas descrições de uma funcionalidade sob a perspetiva do utilizador, focando-se na capacidade desejada e na sua finalidade (Mountain Goat Software, n.d.). A ideia é serem o mais claro e precisas possível, e ajudarem a atingir um nível de detalhe no projeto desenvolvido (Mike Cohn, 2004), podendo ser necessário transformar uma *user story* em várias, caso se queira uma especificação mais detalhada (Mountain Goat Software, n.d.).

Atendendo ao projeto a desenvolver e ao ponto de vista do utilizador final, que neste caso são o profissional de saúde e o doente, foram listadas algumas *user stories*. O objetivo é não só definir as características do *software* como auxiliar também no seu desenvolvimento; deverão ainda ser tidas em conta para a escolha do *hardware* a utilizar, para que este seja o mais adequado possível à finalidade pretendida.

De seguida serão, assim, analisadas todas as *user stories* sob o ponto de vista do doente e do profissional de saúde. Algumas destas *user stories* surgem de sugestões dadas

para este sistema em especial pela *Glintt*, uma empresa com bastante experiência em *software* de saúde e que contacta frequentemente com a classe médica.

3.2.1 Funcionalidades sob a perspetiva do doente

A lista abaixo reflete quais as funcionalidades que o sistema deverá ter sob o ponto de vista do doente como utilizador final.

1. Como doente, quero usar um sistema de monitorização pequeno, confortável e não invasivo para que a minha experiência nas urgências não se torne ainda mais penosa;
2. Como doente, quero que os meus sinais vitais sejam monitorizados para que possa ser detetada atempadamente uma alteração do meu estado de saúde;
3. Como doente, quero que o profissional de saúde responsável tenha acesso à minha localização para que me possa ajudar em caso de necessidade.

3.2.2 Funcionalidades sob a perspetiva do profissional de saúde

De forma análoga, foram listadas as funcionalidades do sistema sob o ponto de vista do profissional de saúde.

1. Como profissional de saúde, quero poder aceder à plataforma do sistema em qualquer local dentro da unidade de saúde para poder ter mobilidade;
2. Como profissional de saúde, quero que o dispositivo de monitorização seja reutilizável e durável, para que esteja sempre disponível quando necessário;
3. Como profissional de saúde, quero que o dispositivo de monitorização a utilizar seja fácil de colocar em funcionamento para não haver demora ou dificuldades no processo;
4. Como profissional de saúde, quero poder inserir um doente no sistema de forma fácil e rápida para acelerar o início da sua monitorização;
5. Como profissional de saúde, quero que a conexão com o dispositivo de monitorização seja simples e rápida para que o sistema entre rapidamente em pleno funcionamento;
6. Como profissional de saúde, quero poder ter acesso a dados do doente, como a idade e o sexo, para poder traçar um perfil;
7. Como profissional de saúde, quero saber qual a situação clínica do doente e qual a sua prioridade, para poder inferir acerca de qual o doente mais suscetível a apresentar alterações nas condições de saúde;
8. Como profissional de saúde, quero ter autonomia para ajustar os parâmetros de alerta dependendo da condição de cada doente individualmente, para que estes sejam adaptados a situações anormais (por exemplo, no caso de o doente ser um atleta ou um idoso, os níveis de alerta não devem ser iguais);
9. Como profissional de saúde, quero ter acesso a uma lista de todos os doentes do sistema para saber, de uma forma geral, quantos doentes estão no sistema e quais necessitam de uma maior atenção;

10. Como profissional de saúde, quero poder ver um mapa da sala com a informação da localização de todos os doentes para poder estar atento à saída de algum doente;
11. Como profissional de saúde, quero ter acesso aos dados de cada doente individualmente para poder verificar a evolução dos seus parâmetros vitais caso ache que o doente em questão precisa de atenção;
12. Como profissional de saúde, quero ter acesso aos parâmetros vitais do doente em tempo real e de forma gráfica para poder verificar o seu estado de saúde e a sua evolução;
13. Como profissional de saúde, quero receber um alerta sempre que um doente apresente alterações notáveis nos seus parâmetros vitais para o poder auxiliar e medicar atempadamente, prevenindo situações de degradação acentuada do estado de saúde ou de morte súbita;
14. Como profissional de saúde, quero que o sistema me alerte quando um doente sai da sala, para detetar que o doente em questão não está a ser devidamente monitorizado e eventualmente possa estar a necessitar de ajuda;
15. Como profissional de saúde, quero poder excluir um doente da lista sempre que a sua passagem pelas urgências termina, para não ter na lista um doente não ativo;
16. Como profissional de saúde, quero poder guardar os dados de um doente que é excluído para poder efetuar uma posterior análise ou simplesmente para registo do episódio clínico.

Tendo em vista as funcionalidades enumeradas anteriormente, prossegue-se, assim, para a definição das especificações técnicas do *software* a implementar.

3.3 Especificações técnicas do *software*

Após a análise dos requisitos impostos e das funcionalidades desejadas, importa estruturar o *software* a realizar, nomeadamente no que diz respeito às interfaces desejadas e às funcionalidades implementadas em cada uma delas. Neste sentido, foram desenhados *mockups*. Importa referir que se optou pelo desenvolvimento de uma aplicação *web*, como explica a Secção 4.2.

Os *mockups* são imagens simples criadas em *software* adequado para este propósito, e que pretendem representar o aspeto da página *web*. Não são imagens interativas, pretendendo apenas comunicar ideias de *design* do *software* a implementar. Através dos *mockups* é possível experimentar cores, *layouts*, fontes, entre outros, de forma rápida, e facilmente usados para obter opiniões e sugestões por parte de outrem antes da implementação da aplicação *web* (Humbrecht, 2016).

A solução a desenvolver será, assim, constituída por quatro interfaces, sendo elas:

1. Interface inicial;
2. Inserir novo doente;
3. Lista de doentes;
4. Informação do doente.

Nas quatro subsecções seguintes encontra-se uma breve descrição do que se pretende em cada interface, bem como uma figura representativa desenhada em *software* específico para o efeito, neste caso o *Balsamiq Mockups 3*. Importa salientar o facto de que estas figuras não correspondem à *dashboard* implementada, sendo apenas uma representação do que se pretende implementar, que no fundo é o que este *software* permite fazer.

3.3.1 Interface inicial

Esta interface será apenas composta pelo logotipo do projeto, uma breve descrição da aplicação e dois botões opcionais para inserir os dados de um novo doente ou para ver a listagem dos doentes que já foram inseridos no sistema. Após a implementação, o aspeto desta interface da *dashboard* deverá ser semelhante ao da Figura 6.



Figura 6 – *Mockup* da interface inicial da *dashboard*.

3.3.2 Inserir novo doente

Esta interface permitirá inserir um novo doente no sistema, tendo apenas de se atribuir um ID (*Identity*) ao doente, que será concedido automaticamente pelo sistema e que não pode ser alterado, e indicar alguns dados, como a idade e o sexo do doente, que permitam auxiliar o processo de localização, caso seja necessário. Assumindo que o algoritmo de localização possa não ser tão preciso quanto seria desejável, estas características irão ajudar a identificar o doente em questão. Para além das informações do doente, esta interface deverá permitir validar a comunicação com a pulseira, por exemplo emitindo uma vibração ou outro sinal semelhante.

De igual forma, esta interface no sistema deverá ser semelhante à da Figura 7.

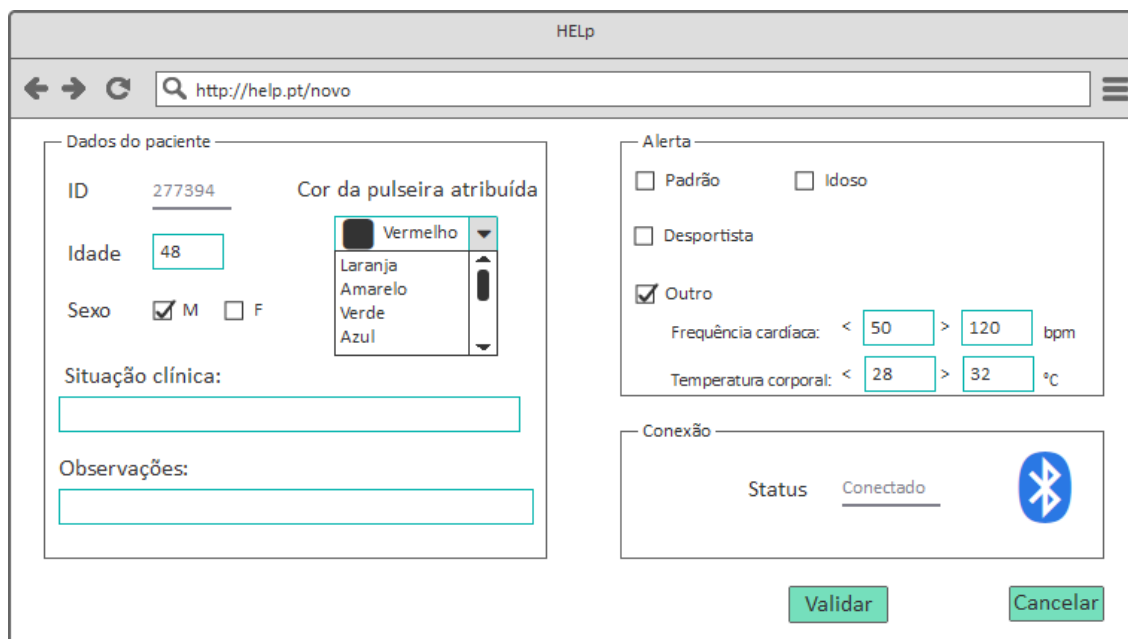


Figura 7 – Mockup da interface que permite inserir um novo doente no sistema.

3.3.3 Lista de doentes

Nesta interface poder-se-á visualizar a listagem de todos os doentes do sistema, nomeadamente o ID do doente, a cor da pulseira que lhe foi atribuída e uma indicação do estado de alarme. Para além disso, poderá eventualmente ver-se a planta do serviço de urgências com indicação da localização de todos os doentes do sistema. A Figura 8 é uma representação desta interface, na qual se pode observar a lista de doentes por ordem de prioridade e um sinal de alerta recebido para um dos doentes da lista. Pode ainda ver-se a data e a hora em que o doente deu entrada, sendo esta correspondente ao momento em que o profissional de saúde validou os dados do doente, sendo por isso automática.

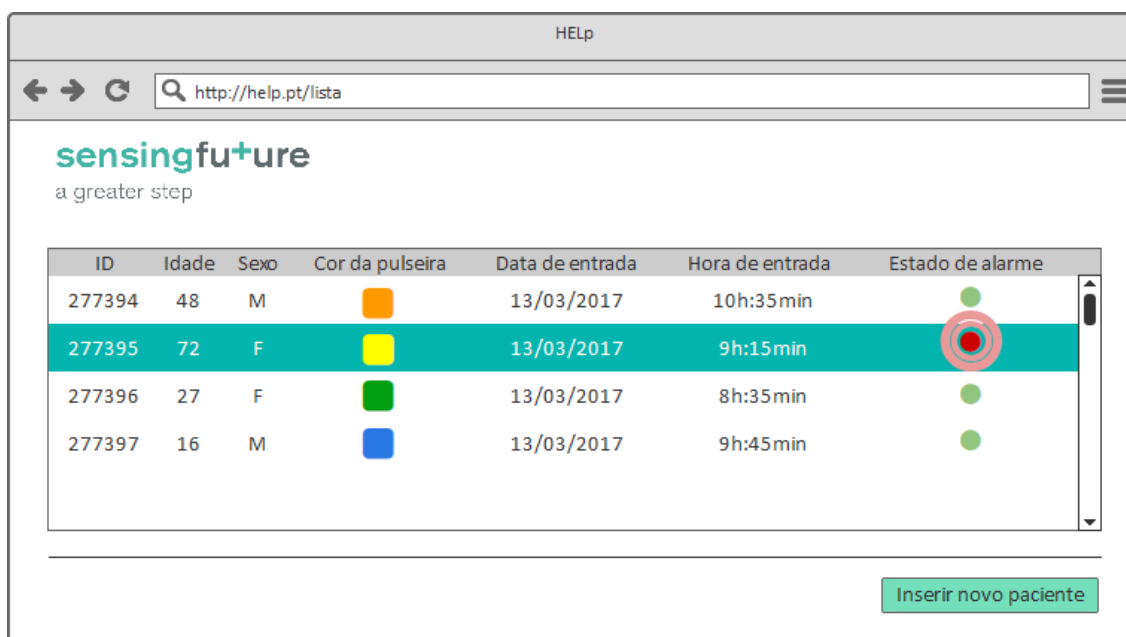


Figura 8 – Mockup da listagem de todos os doentes do sistema com ocorrência de um alerta.

3.3.4 Informação do doente

Tendo sido selecionado um dos doentes da lista apresentada na interface da Figura 8 dever-se-á apresentar os dados do doente em questão, uma indicação da sua localização e do estado de alarme, e um gráfico da variação da frequência cardíaca e da temperatura corporal. Após o doente ter alta médica da urgência, o seu processo poderá ser encerrado, e o doente deixará de poder ser visto na lista. A localização do doente será dada idealmente através de um mapa das urgências, tal como indica a Figura 9.

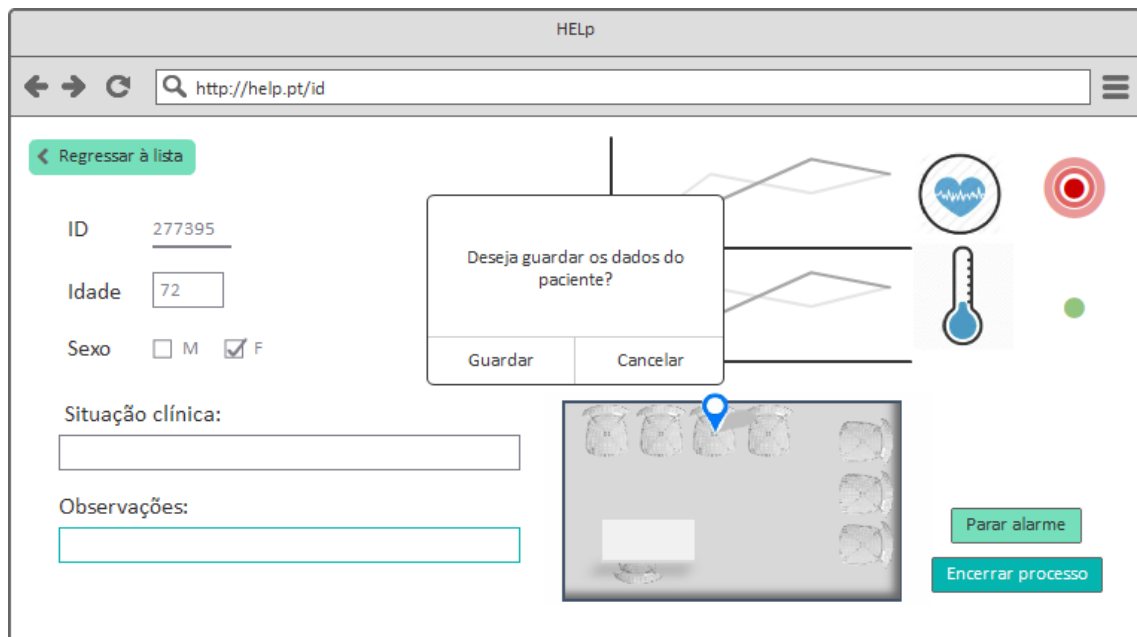


Figura 9 – Mockup da interface com informação do doente e indicação de alerta.

Também neste caso há indicação de um alerta para o doente em questão, que para este exemplo representa uma alteração na frequência cardíaca do doente. Salienta-se, ainda, que ao clicar no botão para encerrar o processo do doente surge uma caixa de texto, e o profissional de saúde pode optar por guardar ou não os dados do episódio de urgência num ficheiro, para posterior avaliação. Após visualizar a informação do doente pode regressar-se à lista de doentes caso não se pretenda ainda encerrar o processo.

Desta forma, será projetado o sistema através da elaboração de esquemas de arquitetura, analisados na Secção 3.5.

3.4 Linguagens de programação

Definido o aspeto da solução de *software* a implementar é essencial proceder à escolha das linguagens de programação a utilizar para a implementação da *dashboard*, bem como qual o ambiente de programação a utilizar. Algumas das linguagens de programação possíveis de utilizar são C/C++, C#, Java, Python, PHP (*Hypertext Preprocessor*), HTML (*Hyper Text*

Markup Language), *JavaScript* e *CSS (Cascading Style Sheets)*. No Capítulo 4 é explicada a escolha das linguagens de programação utilizadas, sendo elas *PHP*, *HTML*, *JavaScript*, *CSS* e *Java*.

De seguida são analisados os esquemas de arquitetura elaborados, nomeadamente diagramas de componentes, de atividades e de sequência, que servirão de base para a implementação do sistema, sendo salientada a sua importância.

3.5 Esquemas de arquitetura

A arquitetura de *software* de um sistema visa a sua estrutura e os componentes que o compõem, permitindo tornar claras as suas características e a forma como se relacionam. Tem como objetivo decompor o sistema em várias partes que se vão construindo de forma iterativa, incrementada e independente. Uma arquitetura de *software* bem documentada é uma mais-valia na medida em que permite explicar a solução e a sua integração no projeto em questão e a fácil compreensão por parte de uma terceira pessoa (Level, Mitra, It, & Ibm, 2008).

Começa-se por analisar o diagrama de componentes (Secção 3.5.1) para se poderem identificar os componentes que farão parte do sistema. É de referir que todos os diagramas foram desenhados através do *software Visual Paradigm*.

3.5.1 Diagrama de componentes

O diagrama de componentes tem como propósito ilustrar como os componentes se relacionam entre si, quais os componentes usados, quais deles podem ser substituídos e quais podem estar em falta. Inclui ainda interfaces e dependências, também estas relacionadas com os restantes elementos (Rosenblum & Science, 2001). Para o sistema a implementar foi assim elaborado um diagrama de componentes, para que se consiga entender quais as interfaces a implementar, os componentes de cada uma e como é feita a interligação entre estes. Este diagrama está ilustrado na Figura 10.

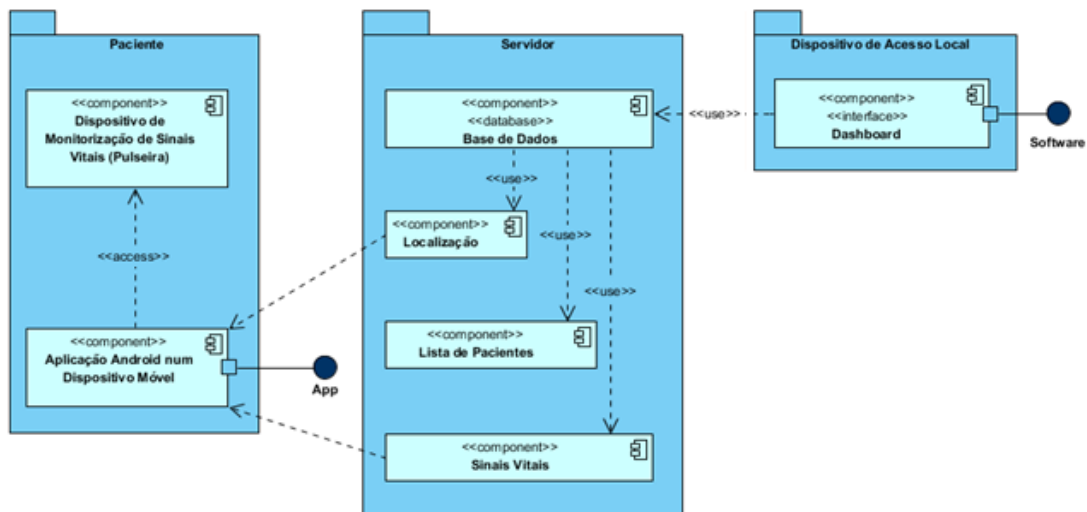


Figura 10 – Diagrama de componentes do sistema a implementar.

O sistema terá, assim, uma componente que diz respeito ao doente e que inclui o dispositivo de monitorização de sinais vitais e uma aplicação *Android* a correr num *smartphone*, necessária à comunicação da pulseira e à recolha de dados a partir desta; um servidor, no qual está implementada uma base de dados onde serão guardados os dados recolhidos pela aplicação *Android*, nomeadamente a localização de cada doente inserido no sistema e os seus sinais vitais; um dispositivo de acesso local, que permite visualizar a *dashboard* do sistema alojada numa plataforma *web*, e que é capaz de apresentar ao utilizador os dados guardados na base de dados.

Desta forma, será analisado de seguida o fluxo de dados que deverá ocorrer na componente de *software*, representados num diagrama de atividades.

3.5.2 Diagrama de atividades

Tal como o diagrama de componentes, o diagrama de atividades pretende descrever a dinâmica do sistema representando o fluxo entre as operações que ocorrem durante a sua utilização. Pretende ainda auxiliar na implementação do sistema analisando a interação entre as diversas atividades (Tutorialspoint, n.d.-d).

O diagrama de atividades do sistema a implementar é o da Figura 11.

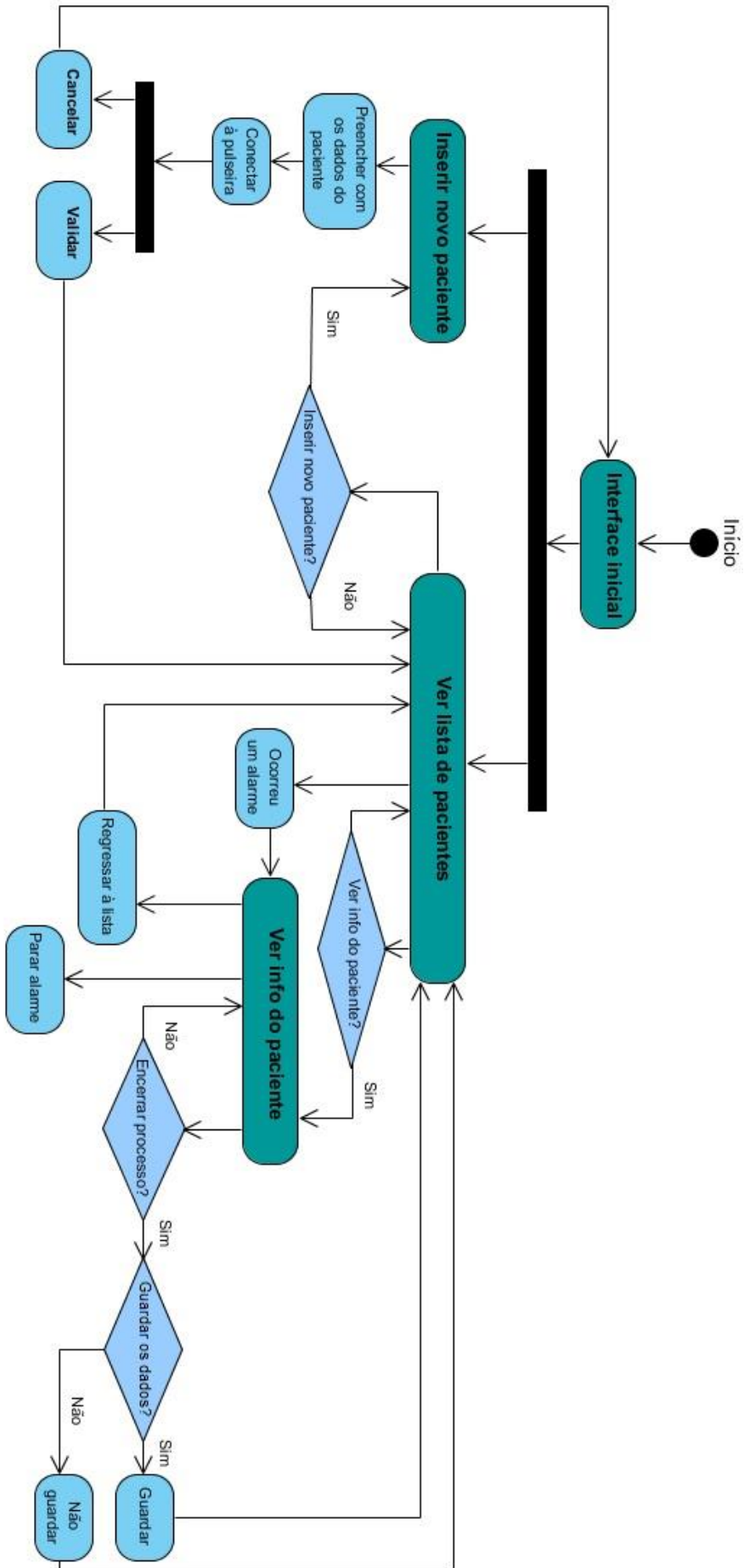


Figura 11 – Diagrama de atividades do sistema a implementar.

Importa referir que deste diagrama fazem parte interfaces, atividades executadas pelo utilizador e eventos do *software*, como é o caso da ocorrência de um estado de alarme. A ação a realizar dependerá das escolhas do utilizador. Caso o utilizador esteja a visualizar a interface inicial, poderá inserir um novo doente no sistema ou visualizar a lista de doentes já inseridos. No primeiro caso, terá de preencher os dados do doente e efetuar a conexão com a pulseira a fim de iniciar a recolha de dados. Este processo apenas estará concluído assim que validado pelo utilizador. Por outro lado poderá optar por visualizar a lista dos doentes no sistema. Esta interface permitirá a qualquer momento inserir um novo doente ou visualizar a informação de qualquer um dos doentes através de um clique, que dará acesso à interface da informação do doente. A partir desta interface o utilizador poderá parar um alarme se for esse o caso, encerrar o processo do doente assim que lhe for dada alta, podendo salvar um documento com os dados do episódio de urgência, ou apenas visualizar os dados e regressar à lista de doentes.

Na Secção 3.5.3 são analisados alguns diagramas de sequência, que representam a sequência de ações que ocorre no sistema para um determinado caso de utilização.

3.5.3 Diagrama de sequência

O diagrama de sequência é usado para mostrar qual a interação entre objetos seguindo uma ordem sequencial, e que geram um determinado resultado que é pretendido. É principalmente usado para expressar os requisitos do sistema ou *use cases* de uma forma mais formal e aprimorada, sendo que cada um dos requisitos pode dar origem a uma ou mais sequências (Bell, 2004).

Para que os pontos-chave deste sistema sejam claramente abordados e expostos, serão apresentados os diagramas de sequência para três das *user stories* indicadas anteriormente, sendo elas: inserir um novo doente no sistema; ter acesso aos dados de cada doente individualmente; receber um alerta sempre que um doente apresenta alterações nos sinais vitais. A Figura 12, a Figura 13 e a Figura 14 retratam os diagramas de sequência obtidos para cada um dos casos.

No primeiro caso (Figura 12) o utilizador pretende inserir um novo doente no sistema. Para isso solicita, através da *dashboard*, a interface “*Inserir doente*”, que lhe permitirá observar um conjunto de campos que tem de preencher com informações relativas ao doente. Terá ainda de estabelecer a comunicação entre a pulseira e a aplicação *Android*, fator decisivo para que o doente possa ser inserido no sistema com êxito. Assim que esta comunicação for estabelecida e for atribuído um ID à pulseira, a aplicação *web* será notificada, e o utilizador poderá validar o processo. Assim que o processo tiver sido validado, a aplicação *web* adicionará uma linha na base de dados, guardando para cada ID de doente os dados inseridos pelo utilizador. Este regressará novamente à interface inicial.

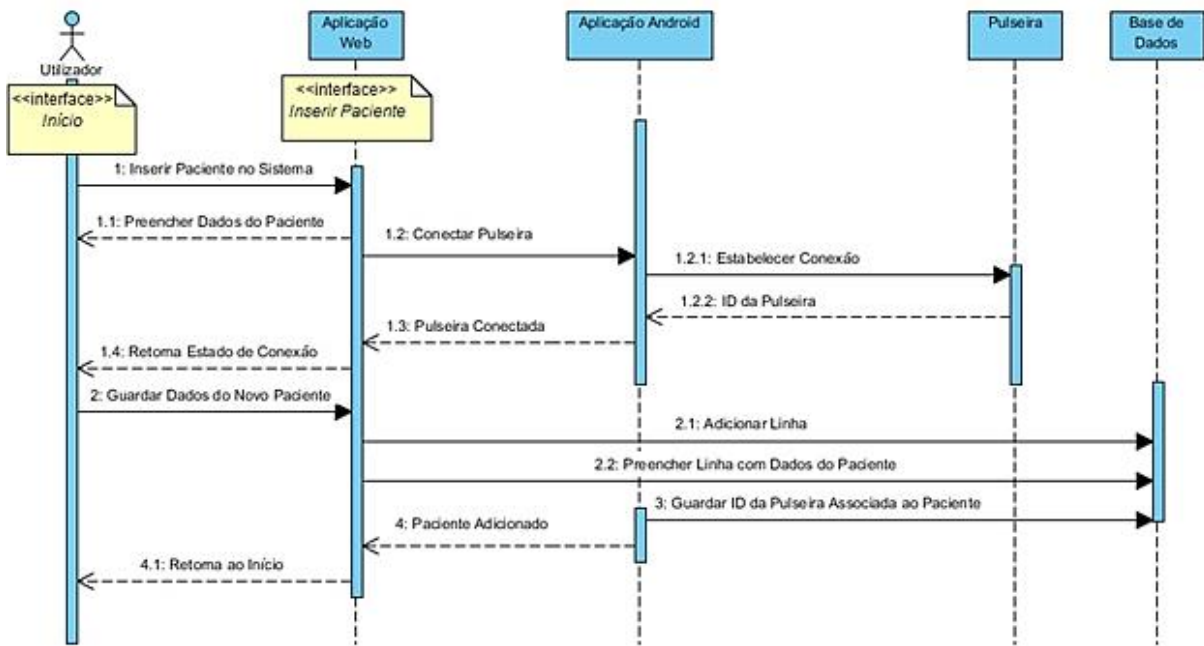


Figura 12 – Diagrama de sequência para o caso “inserir um novo doente no sistema”.

Um outro caso de utilização é a visualização dos dados de um doente da lista, admitindo que o utilizador está a visualizar a lista de todos os doentes inseridos no sistema. Assim, o primeiro passo será seleccionar um dos doentes da lista, pelo que o sistema terá de preencher a interface com os dados guardados na base de dados que dizem respeito ao doente em questão. Admite-se, ainda, que a aplicação *Android* estará continuamente a receber os dados que estão a ser recolhidos pela pulseira e a guardá-los na base de dados, para que possam ser usados assim que necessário e haja uma atualização contínua. A Figura 13 é representativa deste caso de utilização.

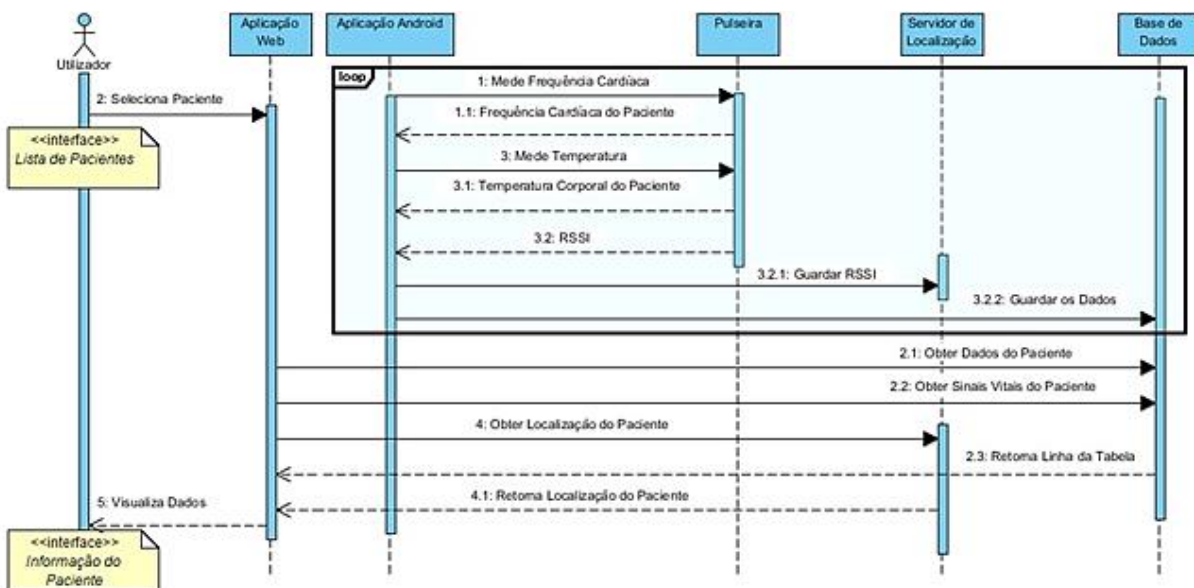


Figura 13 – Diagrama de sequência para o caso “aceder aos dados de um doente individualmente”.

Por fim, outro caso de utilização que é importante analisar é a ocorrência de um alarme. Estando o utilizador a observar a lista de doentes inseridos no sistema, poderá ser notificado da ocorrência de um alarme, podendo assim seleccionar o doente em questão para visualizar os seus dados, tal como no caso anterior, e parar o alarme. Neste caso importa referir que o estado de alarme estará a ser atualizado continuamente através da análise dos sinais vitais do doente, e que o estado de alarme apenas ocorrerá se algum destes sinais apresentar um desvio em relação ao que seria normal. Este caso de utilização está representado na Figura 14.

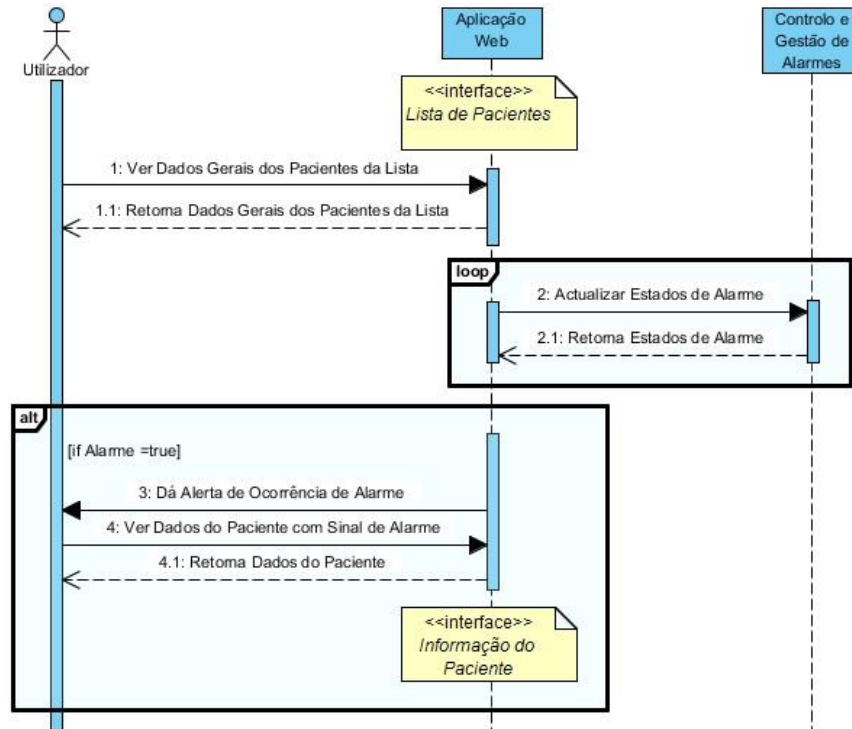


Figura 14 – Diagrama de sequência para o caso de ocorrência de um alarme.

Em suma, os diagramas apresentados permitem entender qual o fluxo de mensagens e respostas sempre que ocorre um evento no sistema. Importa referir que no diagrama da Figura 14 foram omissos os pedidos de informação do doente à base de dados e à aplicação, bem como ao dispositivo de monitorização, uma vez que estes passos já se encontravam no diagrama da Figura 13. É ainda de reter que a existência de ciclos correspondentes às atualizações do sistema são desejáveis num sistema deste tipo, para que se esteja constantemente a adquirir e guardar sinais vitais, ou a atualizar os estados de alerta para que o utilizador seja notificado e possa agir de forma imediata.

Terminada a análise de especificações técnicas do *software* podem analisar-se as especificações técnicas do *hardware*, explicadas na Secção 3.6

3.6 Especificações técnicas do hardware

De acordo com a solução pretendida, foi feita uma pesquisa alargada para encontrar um dispositivo, fosse ele uma pulseira ou qualquer outro tipo de solução, capaz de monitorizar a frequência cardíaca e a temperatura corporal, parâmetros necessários a este sistema. Deveriam ainda implementar um protocolo de comunicação sem fios, podendo ser *Bluetooth* ou *Wi-Fi*. Várias soluções foram de encontro ao que era pretendido, tendo no entanto como maior desvantagem o preço, que em certos casos não foi possível conferir.

Algumas das soluções encontradas com toda a informação desejada e que vão de encontro aos objetivos pretendidos encontram-se expostas na Tabela 6. Importa referir que a comunicação em todas elas é estabelecida através do protocolo BLE. Nesta tabela são ainda apresentadas outras funcionalidades que são comuns a várias soluções, ainda que não sejam relevantes para a escolha do dispositivo a utilizar, destacando novamente que todas estas soluções são capazes de medir a frequência cardíaca e a temperatura corporal, e apresentam capacidade de conexão *Bluetooth*.

Tabela 6 – Possíveis soluções de monitorização encontradas, capazes de medir os sinais vitais desejados e de conexão *Bluetooth*.

Designação	Marca	SpO ₂	Taxa respiratória	Monitor de atividade	Padrão de sono	Notificações de SMS ou redes sociais
Pulseiras						
<i>VinCense</i>	<i>VinCense</i>	✓	✗	✗	✗	✗
<i>Microsoft Band 2</i>	<i>Microsoft</i>	✗	✗	✓	✓	✓
<i>Blocks</i>	<i>BLOCKS</i>	0	0	0	0	0
<i>E4 Wristband</i>	<i>empatica</i>	✗	✗	✓	✗	✗
<i>Amiigo</i>	<i>Amiigo</i>	✓	✓	✓	✓	✗
<i>Hesvitband S3</i>	<i>Hesvit</i>	✗	✗	✓	✓	✗
<i>UP3</i>	<i>Jawbone</i>	✗	✗	✓	✓	✓
<i>metahealth</i>	<i>mbientlab</i>	✗	✗	✓	✗	✗
<i>X7 Wristband</i>	--	✗	✗	✓	✓	✓
<i>H6 Smart Bracelet</i>	--	✗	✗	✓	✓	✓
<i>Heroyu Smart Bracelet</i>	--	✓	✗	✓	✓	✓
Pensos						
<i>VitalPatch</i>	<i>VitalConnect</i>	✗	✓	✓	✗	✗
Fitas para o cabelo						
<i>Spree Headband</i>	<i>Spree Wearables</i>	✗	✗	✓	✗	✗
<i>iBand +</i>	<i>Arenar</i>	✗	✗	✓	✓	✗
Anéis						
<i>Oura Ring</i>	<i>Oura</i>	✗	✓	✓	✓	✗
Auriculares						
<i>Cosinuss One</i>	<i>Cosinuss</i>	✗	✗	✗	✗	✗

✓ – Apresenta esta funcionalidade

✗ – Não apresenta esta funcionalidade

0 – Não aplicável.

Importa referir que a *Sensing Future Technologies* não pretendia desenvolver um protótipo de *hardware* de raiz, mas sim utilizar um dispositivo comercial que oferecesse algumas das funcionalidades pretendidas, de forma a integrá-lo no protótipo desenvolvido, o que explica toda a pesquisa realizada.

Através da análise da Tabela 6, pode verificar-se que a característica mais comum para a maioria dos dispositivos elegidos é a monitorização da atividade, o que salienta o facto de estes dispositivos serem na sua maioria monitores de *fitness* e não dispositivos médicos. Os típicos dispositivos de desporto têm ainda a funcionalidade de emitirem uma notificação quando a pessoa recebe uma chamada ou SMS (caso estejam a uma distância do *smartphone* aceitável e que permita o envio e receção de dados através do *Bluetooth*, pelo qual se encontram conectados) ou quando é recebida uma notificação das redes sociais. Nenhuma destas funcionalidades é relevante para o sistema em questão; apenas se salienta a diferença entre dispositivos relativamente à sua utilização mais comum.

3.7 Considerações finais

A aplicação de métodos de desenvolvimento de *software*, nomeadamente no que diz respeito à identificação dos requisitos e da arquitetura do sistema, revela-se bastante útil. Em primeiro lugar é importante refletir sobre quais as principais características que o sistema deverá ter de acordo com o objetivo proposto. De seguida, é essencial fazer um levantamento de quais as funcionalidades desejáveis para o utilizador final, tentando corresponder a todas na implementação. No fundo, estas funcionalidades representam etapas na implementação do *software*, sendo satisfeitas uma a uma. É ainda de referir que, apesar de numa fase inicial, estas funcionalidades terem sido pensadas pela aluna, em alguns casos foram adaptadas para ir de encontro aos requisitos dados pela empresa *Glintt*, como referido anteriormente. No final, e atendendo às funcionalidades enumeradas, os *mockups* constituem um “rascunho” da *dashboard* a implementar, permitindo estudar principalmente o *design* que melhor se adapta à aplicação, tornando-a o mais apelativa possível. Por sua vez, os diagramas de componentes, de atividades e de sequência permitem condensar o funcionamento do sistema, identificando todos os componentes necessários, representando os passos a seguir ao utilizar a *dashboard*, dando a entender o seu comportamento perante determinada ação por parte do utilizador e exibindo a interação entre os seus componentes, ajudando também na sua implementação.

Por fim, é de salientar o vasto leque de novas competências adquiridas, nomeadamente ao nível dos métodos de desenvolvimento de *software* utilizados, os *softwares Balsamiq Mockups 3* e *Visual Paradigm*, o que contribuiu para expandir os conhecimentos da aluna na área do desenvolvimento de *software*. Relativamente ao *hardware*, é de referir que a maior parte dos dispositivos encontrados são utilizados para aplicações de *fitness*, o que explica o facto de as suas funcionalidades irem de encontro a atividades do dia-a-dia e os sinais vitais que permitem medir serem escassos. O preço é ainda um fator limitante, uma vez que sendo elevado torna a aquisição do sistema insuportável. Algumas das soluções encontradas mais dirigidas para a área hospitalar representavam ainda protótipos, não disponíveis para compra, pelo que não puderam ser consideradas como possível *hardware* a ser usado neste projeto.

Salienta-se assim a dificuldade de encontrar um dispositivo comercial capaz de satisfazer todas as necessidades do projeto a um custo reduzido.

4. Descrição da solução implementada

Neste capítulo é feita a descrição do sistema desenvolvido, sendo apresentadas todas as etapas do processo de desenvolvimento, desde o seu início até à conclusão do protótipo, sob a forma de uma linha temporal na Secção 4.1.

De seguida, na Secção 4.2, é explicada a seleção de *software*, fazendo-se uma breve abordagem às principais características das *frameworks*, linguagens e bibliotecas selecionadas.

De forma semelhante, o processo de seleção de *hardware* é descrito na Secção 4.3, sendo apresentados os dispositivos selecionados e a forma como foram avaliados (Secção 4.3.1), indicando ainda algumas aplicações testadas.

Na Secção 4.4 é feita uma abordagem à aplicação *Android* desenvolvida com o objetivo de recolher os dados dos parâmetros vitais do doente (Secção 4.4.1), expondo o método de *fingerprinting* na Secção 4.4.2.1, aliado à tecnologia *Wi-Fi*, bem como a integração da localização *indoor* (Secção 4.4.2.2) nesta aplicação.

De seguida é exposto o processo de implementação da *dashboard* (Secção 4.5), dando-se destaque à conceção da base de dados (Secção 4.5.1), e a todas as etapas do seu desenvolvimento, incluindo o desenvolvimento das interfaces e a interação entre as interfaces e a base de dados (Secção 4.5.2) e ações responsivas (Secção 4.5.3).

Na Secção 4.6, é apresentada a comunicação entre a aplicação *Android* e a base de dados, sendo abordados dois protocolos de comunicação: REST API (Secção 4.6.1) e SOAP (Secção 4.6.2). Faz-se ainda uma comparação entre estes (Secção 4.6.3), indicando-se na Secção 4.6.4 a forma como foram utilizados.

Por fim, na Secção 4.7, são apresentadas algumas considerações finais acerca do trabalho desenvolvido e abordado neste capítulo, dando-se destaque às tecnologias usadas e aos desafios encontrados.

4.1 Etapas do processo de desenvolvimento

A Figura 15 representa uma linha temporal com indicação do fluxo das principais etapas do desenvolvimento do protótipo. É importante referir que, tal como se pode verificar, algumas etapas foram realizadas em simultâneo, pretendendo-se ainda dar uma ideia do tempo dedicado a cada uma delas.

Aquando do desenvolvimento da solução, houve alguns aspetos, definidos na fase do planeamento, que foram ajustados, com vista à melhoria no tempo de preenchimento de dados e à facilidade de utilização do *software*. Alguns destes aspetos dizem respeito ao *design* e a algumas funcionalidades, sendo abordados na Secção 4.5.2.

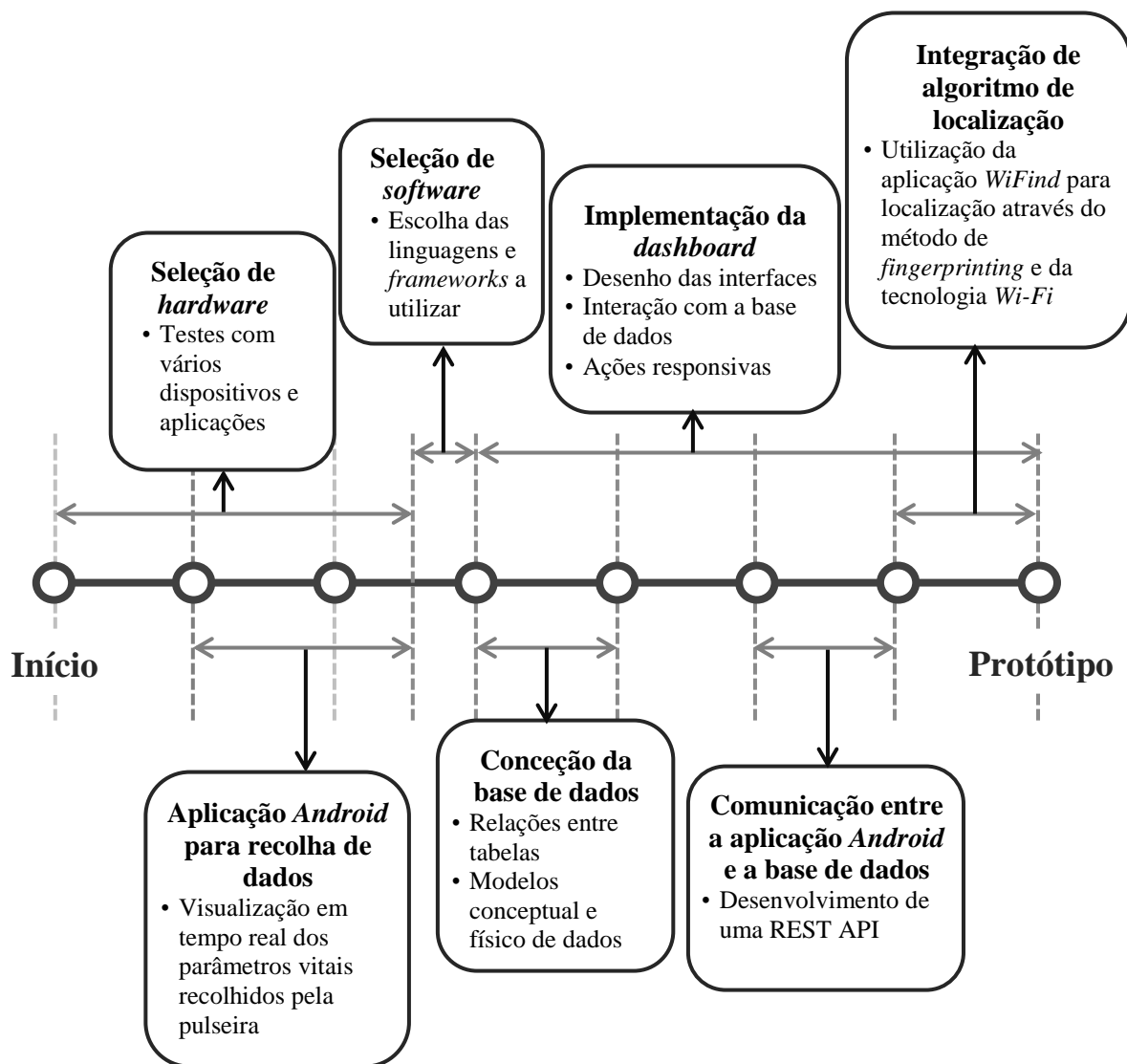


Figura 15 – Fluxo de tarefas durante o desenvolvimento do sistema.

Relativamente à Figura 15, é de salientar que o processo de seleção de *hardware* se revelou mais complexo do que inicialmente se pensava, como se pode verificar na Secção 4.3. Também a implementação da *dashboard* ocupou bastante tempo de desenvolvimento, englobando o planeamento dos requisitos e arquitetura do sistema anteriormente descritos no Capítulo 3, a implementação da base de dados (Secção 4.5.1) e os protocolos de comunicação entre a aplicação *Android* e a base de dados para guardar os dados dos sinais vitais do doente e a sua localização, como é explicado na Secção 4.6.

4.2 Seleção de software

Atendendo ao facto de que foi necessário implementar uma aplicação *Android* capaz de comunicar com o dispositivo de monitorização de sinais vitais e recolher dados, a linguagem de programação usada foi a linguagem *Java*, suportada pelo *Integrated Development Environment (IDE) Android Studio*, sendo este o mais comum no desenvolvimento de aplicações *Android*.

Relativamente ao *software* a utilizar para a implementação da *dashboard*, foram analisadas as várias linguagens sugeridas na Secção 3.4. Optou-se pela linguagem PHP e pela *framework Laravel* por ser uma ferramenta já utilizada na empresa de acolhimento deste estágio para outras aplicações e, portanto, já conhecida. Pretendeu-se também oferecer portabilidade, podendo a aplicação funcionar em diferentes sistemas operativos por implementar uma plataforma *web*, características que não seriam possíveis de alcançar através do uso das restantes linguagens sugeridas. Adicionalmente foi necessário adquirir novos conhecimentos, nomeadamente em conceitos de HTML, CSS, *JavaScript*, *jQuery*, *Ajax* e *Bootstrap*. A construção da base de dados foi feita recorrendo ao *MySQL*.

Uma vez decidido qual a linguagem de programação a usar e quais as *frameworks* e linguagens adicionais a utilizar, foi relevante efetuar uma breve pesquisa acerca de cada uma delas com o objetivo de nos inteirarmos sobre as principais funcionalidades de cada uma delas quando utilizadas numa interface *web*. O resultado deste trabalho é apresentado nas duas subsecções seguintes.

4.2.1 Linguagem de programação PHP

A programação com recurso à linguagem PHP permite obter uma característica importante para sistemas deste tipo - portabilidade.

De acordo com o manual de PHP presente na página oficial (www.php.net), esta é uma linguagem *open source* largamente utilizada no desenvolvimento de aplicações *web*, sendo usada por programadores com o principal objetivo de criar páginas *web* de forma fácil e dinâmica. A sua sintaxe é baseada em C e em *Java*, o que a torna simples de aprender, e pode ser embutida em HTML (Achour et al., 2017). Para acelerar o processo de desenvolvimento e poder criar uma aplicação mais estável e disponível, o uso de *frameworks* para PHP é essencial (J. Reyes, 2009).

O uso de bibliotecas permite auxiliar o programador a reduzir significativamente o tempo de desenvolvimento e a tornar esta tarefa de desenvolvimento de *software* mais interessante, permitindo focar-se mais noutros aspetos, como o *design* da página *web* (Sharma, 2015). Na Secção 4.2.2 é feita uma abordagem às *frameworks* e às bibliotecas a utilizar neste projeto.

4.2.2 *Frameworks* e bibliotecas

Numa primeira fase importa esclarecer a diferença entre uma *framework* e uma biblioteca, dado que ambas ajudam a facilitar o processo de desenvolvimento, servindo portanto o mesmo propósito. Uma *framework* permite definir a estrutura dos dados onde a aplicação define as suas funcionalidades, sendo que o código é chamado pela *framework* sempre que necessário. Por sua vez, uma biblioteca contém apenas definições de classes e métodos que

podem ser reutilizadas por outros programadores. Isto permite que o programador possa usar código já existente na estrutura da sua aplicação criada através da *framework* (Shubham Kumar, 2015).

As *frameworks* utilizadas foram o *Laravel*, pelo mesmo motivo que levou à escolha da linguagem PHP, e a *framework Bootstrap*. Adicionalmente foi necessário o uso das bibliotecas HTML e CSS, *JavaScript*, *jQuery* e *Ajax*. Nas secções 4.2.2.1 a 4.2.1.6 é feita uma breve abordagem ao *Laravel*, ao *Bootstrap* e a todas as bibliotecas mencionadas nas secções seguintes.

4.2.2.1 *Laravel*

De acordo com a página oficial do *Laravel* (www.Laravel.com), esta *web framework*, de sintaxe simples e expressiva, permite uma experiência de desenvolvimento agradável e criativa pela sua facilidade de uso (*Laravel*). É uma *framework open source* que possibilita várias funcionalidades, nomeadamente modularidade, testabilidade e roteamento (*Tutorialspoint*).

As vantagens da utilização do *Laravel* incluem a simples implementação de autenticação de sistemas, a integração com serviços de correio eletrónico para envio de notificações, a proteção contra vulnerabilidades técnicas comuns, um suporte para teste e verificação, e a separação entre código lógico e de apresentação, sendo o *Laravel* uma *framework Model View Controller* (MVC) (Kazankov, 2016).

4.2.2.2 *Bootstrap*

Bootstrap é uma *framework* CSS que disponibiliza um alargado número de opções de estilos base para páginas *web* (*Caelum*). Usa ainda HTML e *JavaScript* (abordadas de seguida, assim como CSS), pelo que é necessário conhecer alguns conceitos sobre estas linguagens para que possam ser utilizados. As vantagens da sua utilização incluem o seu suporte por parte dos *browsers* que são usados atualmente, a facilidade de uso devido à documentação disponível e ao *design* responsivo, providenciando uma solução simples e funcional (*Tutorialspoint*). O seu *design* responsivo deve-se ao sistema de grelhas do *Bootstrap*, no qual existem 12 colunas individuais que são rearranjadas de acordo com o tamanho do ecrã do dispositivo utilizado (“*Bootstrap Grid System*,” n.d.).

4.2.2.3 HTML e CSS

HTML é a linguagem de marcação usada na escrita de conteúdos de páginas *web*. É composta por *tags*, escritas aos pares, entre as quais é disposto o código. São estas *tags* que permitem agrupar o código que é traduzido pela página *web* para conteúdo que pode ser lido (Boswell, 2016).

As linguagens HTML e PHP apresentam capacidade de interação entre si, sendo que PHP contém código HTML embebido, o que permite, quando se está a programar em HTML, “saltar” para código PHP (Achour et al., 2017). Uma outra vantagem inclui poder ter código centralizado num único local, que pode ser usado por vários elementos da página, como é exemplo a criação de um cabeçalho único para determinado *site* que é aproveitado em todas as suas páginas. Com PHP é possível criar o cabeçalho num ficheiro .php e posteriormente incluir esse ficheiro em todas as páginas usando um único comando (Caelum).

CSS (*Cascading Style Sheets*) é uma linguagem simples de *design* usada para controlar as cores do texto da página *web*, os estilos e fontes, o espaço entre parágrafos, etc. Permite ter controlo sobre a apresentação da página HTML, linguagem com a qual é tipicamente associada (Tutorialspoint). Por sua vez, a linguagem CSS pode interagir com HTML através de PHP (CSS Newbies, 2015), pelo que ser utilizado neste projeto constitui uma vantagem.

4.2.2.4 *JavaScript*

A linguagem *JavaScript* é a responsável pela dinâmica das páginas *web*, sendo atualmente suportada por todos os navegadores. É uma linguagem de *scripting*, permitindo o controlo de aplicações terceiras, como é o caso do HTML. As suas vantagens incluem a independência de compilação para que o código possa ser executado (Caelum). Não deve ser confundida com *Java*. No caso deste trabalho em particular, a linguagem *JavaScript* pode ser usada na validação de formulários e na elaboração de gráficos, entre outros.

4.2.2.5 *jQuery*

jQuery é uma biblioteca de *JavaScript* que tem como objetivo simplificar a manipulação e animação de documentos HTML pela escrita simplificada de código. Esta biblioteca possibilita o uso de todas as funções e capacidades de *JavaScript*, e permite ainda aproveitar os seletores CSS para permitir um rápido acesso a elementos (Tutorialspoint).

4.2.2.6 *Ajax*

Ajax é um conjunto de ferramentas *JavaScript* e *eXtensible Markup Language* (XML) que permite a comunicação entre o *script* do servidor e determinado objeto sem que haja necessidade de atualizar toda a página *web*: apenas é atualizado o objeto em questão (Mozilla Developer Network and individual contributors, 2017). Constitui, portanto, uma mais-valia para diversas aplicações, incluindo o projeto desenvolvido.





Para além da decisão sobre qual o *software* a utilizar, foi essencial a escolha de um *hardware* com características e capacidades adequadas à função pretendida, pelo que na Secção 4.3 são analisados vários dispositivos que foram testados e avaliados.

4.3 Seleção de *hardware*

Devido à maior oferta e acessibilidade, decidiu-se usar uma pulseira ao invés de qualquer um dos outros tipos de dispositivos discutidos na Secção 3.6. É ainda de salientar que usar *hardware standard* e comercial foi um dos pressupostos definidos para este projeto pela empresa, uma vez que oferece vantagens como a poupança no tempo de desenvolvimento e a redução do preço devido à produção em massa. Para além das vantagens mencionadas, acredita-se que este tipo de dispositivos já foram devidamente testados e foram sujeitos a testes de qualidade, nomeadamente de biocompatibilidade, pelo que se encontram assim em condições de serem usados.

A Tabela 7 indica quais as pulseiras usadas para testes pela aluna, indicando também a aplicação *Android* disponibilizada pela marca de cada uma.

Tabela 7 – Pulseiras utilizadas no desenvolvimento do sistema.

	Designação	Fabricante	Aplicação <i>Android</i>
	<i>Mi Band 1S</i>	<i>Xiaomi</i>	<i>Mi Fit</i>
	<i>X7 Wristband</i>	--	<i>GetFit 2.0</i>
	<i>UP3</i>	<i>Jawbone</i>	<i>UP</i>
	<i>Microsoft Band 2</i>	<i>Microsoft</i>	<i>Microsoft Band</i>

Na fase inicial usou-se a pulseira *Mi Band 1S*, que não se encontra listada na Tabela 6 apresentada na Secção 3.6. Na realidade, esta pulseira não vai de encontro aos pressupostos, uma vez que não tem capacidade de monitorização da temperatura corporal. No entanto, mede

a frequência cardíaca e é capaz de comunicar com um *smartphone* através do *Bluetooth*, pelo que se considerou adequada para uma primeira fase de experiências e testes. Já existia uma destas pulseiras na empresa, disponível para ser usada de imediato, outro fator decisivo para a sua utilização. Posteriormente, foi adquirida a pulseira *X7 Wristband*, por ser a mais barata da lista. Também com esta foram realizados testes. Foram também utilizadas a *UP3* e a *Microsoft Band 2*.

O objetivo da utilização de todos estes dispositivos será explicado na Secção 4.2.1, sendo ainda apresentadas as razões pelas quais as pulseiras *Mi Band 1S*, *X7 Wristband* e a *UP3* foram rejeitadas, não fazendo parte do protótipo final e as razões pelas quais a *Microsoft Band 2* foi escolhida. Importa ainda referir que, para todas elas, foi explorada a aplicação oficial disponível na *Play Store* do *Google*.

4.3.1 Avaliação e seleção de *hardware*

O primeiro passo no desenvolvimento deste sistema consistiu na tentativa de utilização de um dos dispositivos de monitorização disponíveis, neste caso a *Mi Band 1S*. O objetivo inicial estabelecido sugeria tentar encontrar-se uma aplicação *open source* capaz de se conectar à *Mi Band 1S*, com o objetivo de tentar perceber como é que esta comunica com um *smartphone* e como se poderiam adquirir os sinais medidos pela pulseira. Esta pesquisa permitiu encontrar algumas aplicações desenvolvidas por programadores amadores; no entanto, devido à sua dimensão e à dificuldade de entender os exemplos de código encontrado, optou-se por desenvolver uma aplicação própria.

Esta aplicação foi desenvolvida utilizando o IDE *Eclipse*, uma ferramenta que permite o desenvolvimento de aplicações *Android* em linguagem *Java*. Usando esta aplicação é possível detetar dispositivos *Bluetooth* (independentemente de serem *Bluetooth* ou BLE) nas imediações do dispositivo móvel onde se corre a aplicação. Estes dispositivos são listados, sendo indicado o nome, endereço *Bluetooth* e RSSI de cada um. Quando o utilizador seleciona um dispositivo, a aplicação identifica qual o protocolo de comunicação e tenta estabelecer uma conexão. Na Figura 16 é possível ver a interface principal da aplicação desenvolvida: na Figura 16 (a) pode observar-se que o *Bluetooth* se encontra desligado; clicando no botão “*Connect*” e após dado o consentimento por parte do utilizador, o *Bluetooth* passa a estar ligado e é iniciada a descoberta de dispositivos *Bluetooth* próximos do *smartphone* que está a ser usado, como mostra a Figura 16 (b).

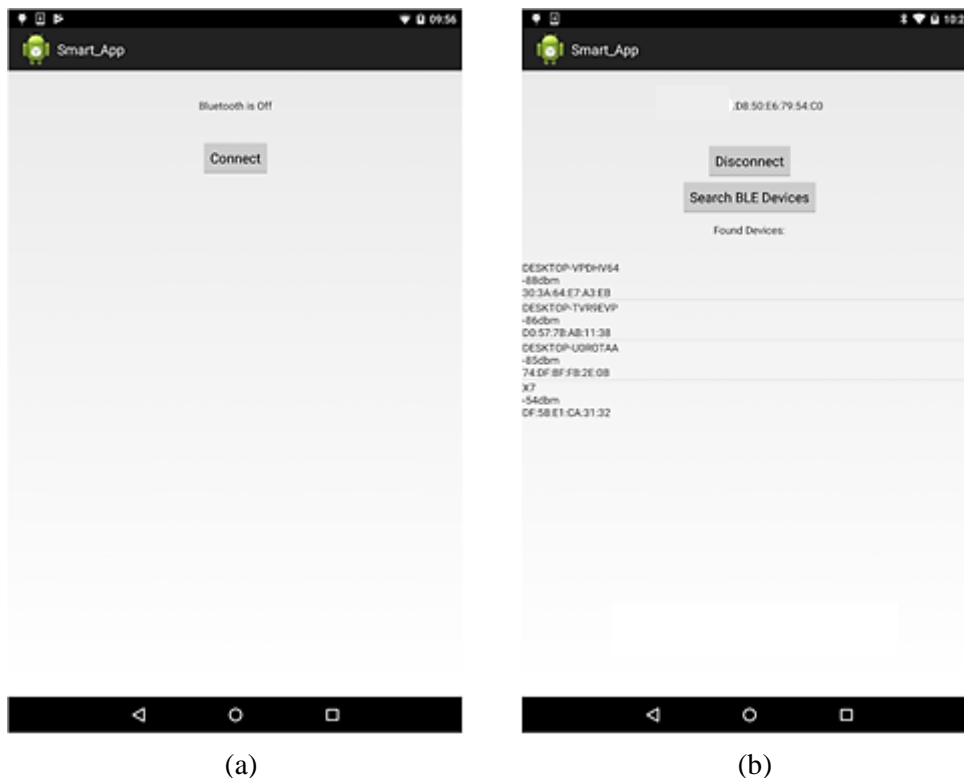


Figura 16 – Aplicação *Android* desenvolvida para conexão com as pulseiras de teste. (a) Interface inicial da aplicação, com *Bluetooth* desligado. (b) Descoberta de dispositivos *Bluetooth* nas proximidades, com indicação do nome, endereço *Bluetooth* e RSSI dos dispositivos encontrados.

Esta aplicação funciona para dispositivos *Bluetooth* clássicos, como telemóveis e computadores; quando tenta estabelecer uma ligação com a *Mi Band 1S*, no entanto, esta conexão é rejeitada pela pulseira, tornando assim impossível a recolha de dados. O mesmo aconteceu quando se tentou conectar com a *UP3*. O motivo desta falha pode estar associado à segurança conferida pelo *firmware* do dispositivo, que impede a recolha de dados por parte de aplicações não oficiais.

Tentou-se, assim, usar esta aplicação para a conexão com a *X7 Wristband*. Para este modelo foi possível estabelecer a conexão; porém, dado ser uma pulseira de marca desconhecida e não existirem quaisquer projetos *open source* para esta pulseira em especial, não são conhecidos os *Universally Unique Identifiers* (UUID's) das características necessárias ao envio e receção de informação. A descoberta destes UUID's e a associação ao serviço *Bluetooth* correspondente não seria possível de realizar em tempo útil, atendendo à data de término do estágio, pelo que se decidiu colocar esta pulseira de parte e explorar outras opções.

O objetivo desta nova pesquisa de soluções incidu, assim, na procura de novas pulseiras cuja marca disponibilizasse uma *Application Programming Interface* (API) que permitisse a programação por parte de amadores ou, sob outro ponto de vista, na procura das denominadas “*Third Party Applications*” ou de plataformas *online* de recolha de dados, ambas capazes de recolher informação a partir das aplicações *Android* para as pulseiras existentes, como são exemplo as da Tabela 7. Relativamente a esta possível solução, verificou-se que

existem várias plataformas, mas é impossível recolher os dados com uma periodicidade adequada com algumas delas, como é o caso da *UP3*. No entanto, verificou-se que várias marcas disponibilizam uma API gratuita para desenvolvimento, como é o caso da *Microsoft*. Foi ainda possível obter esta API e o respetivo manual de utilização através do contacto com utilizadores externos, uma vez que a marca já não os disponibiliza. A Tabela 8 apresenta, para cada uma destas pulseiras, outras soluções utilizadas, bem como quais as dificuldades encontradas e que levaram a que tenham sido colocadas de parte.

Tabela 8 – Algumas aplicações exploradas para cada uma das pulseiras utilizadas.

Pulseira	Aplicação	Tipo	Dificuldades Encontradas
<i>Mi Band 1S</i>	<i>Google Fit</i>	Plataforma <i>online</i>	Não foi possível exportar os dados da frequência cardíaca.
	<i>Notify & Fitness for Mi Band</i>	<i>Third Party Application</i>	Sincronização com o <i>Google Fit</i> tem custos, e não se conseguiu apurar se é possível exportar os dados da frequência cardíaca.
<i>X7 Wristband</i>	<i>nRF Connect</i>	Aplicação <i>Android</i>	Foi usada para tentar descobrir os <i>UUID's</i> das características. Realmente surgiram alguns <i>UUID's</i> , mas não identificados.
<i>UP3</i>	Portal <i>Jawbone</i>	Site <i>online</i>	Não permitiu aceder à frequência cardíaca em tempo real.
	<i>FitnessSyncer</i>	Plataforma <i>online</i>	Não foi possível exportar os dados da frequência cardíaca.
	<i>IFTTT</i>	Plataforma <i>online</i>	Não foi possível exportar os dados da frequência cardíaca.

Dado o facto de se poder usar o SDK da *Microsoft Band 2*, esta surgiu como uma escolha bastante plausível. Existia ainda uma pulseira desta marca e modelo disponível no Laboratório de Automática e Sistemas do Instituto Pedro Nunes, pelo que se decidiu realizar uma aplicação de teste para este dispositivo.

A implementação desta aplicação apresentou-se no início como um desafio, pois teria de ser utilizada com o suporte do *Visual Studio* e da linguagem *C#*, pouco habitual para aplicações *Android*. No entanto, após alguma pesquisa, verificou-se que podia ser usado o *Android Studio* para esta implementação e sem que o SDK fosse necessário, podendo apenas usar-se uma biblioteca da *Microsoft Band* como suporte, facilmente encontrada na *Internet*. Procedeu-se então à implementação de uma aplicação funcional, capaz de recolher os dados da frequência cardíaca e da temperatura corporal medidos pela pulseira em tempo real.

4.4 Aplicação *Android* de recolha de dados

O desenvolvimento da aplicação *Android* necessária ao funcionamento do sistema de monitorização foi repartido em três etapas principais, sendo elas:

1. Desenvolvimento de uma aplicação *Android* para exibir, em tempo real, os valores da frequência cardíaca e da temperatura corporal medidos pela pulseira;
2. Integração do algoritmo de localização *indoor*;
3. Capacidade de envio dos dados relativos aos parâmetros vitais e à localização de um doente para a base de dados do sistema, associados a um episódio de urgência.

Para cada uma destas etapas é exposto, nas subsecções seguintes, qual o procedimento realizado.

4.4.1 Desenvolvimento de uma aplicação *Android* para a *Microsoft Band 2*

Como já referido, foi implementada uma aplicação *Android* capaz de comunicar com a *Microsoft Band 2* e de mostrar, em tempo real, os dados da frequência cardíaca e da temperatura corporal recolhidos pela pulseira. Outro objetivo desta aplicação era a recolha do RSSI da pulseira em tempo real, que iria fazer parte do módulo de localização.

Para isso foi desenvolvido um projeto com recurso ao *software Android Studio*, sendo apenas necessário importar para o projeto uma biblioteca da *Microsoft*, que contém código fonte para comunicação com esta pulseira em especial. Recorrendo à documentação da SDK disponibilizada pela *Microsoft* e a alguns projetos do *GitHub*, foi conseguida uma primeira versão desta aplicação, cuja interface é a da Figura 17.

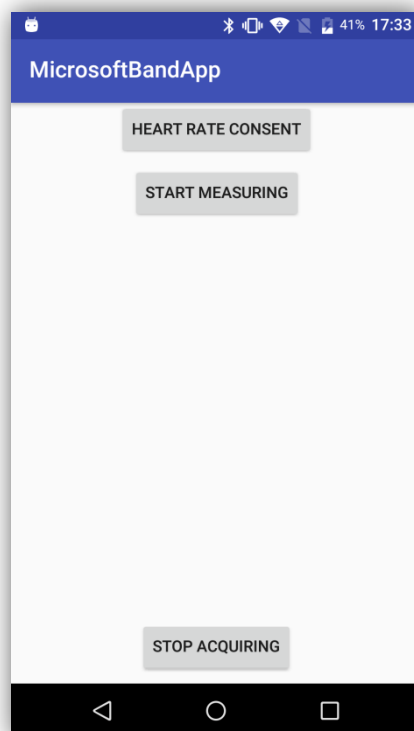


Figura 17 – Interface da primeira versão da aplicação *MicrosoftBandApp*.

Importa referir que a aplicação não faz a descoberta de dispositivos *Bluetooth* na proximidade do *smartphone* utilizado, nem a tentativa de emparelhamento, partindo do princípio de que a pulseira já se encontra emparelhada. Internamente é feita uma descoberta ao dispositivo emparelhado e verificado o estado de emparelhamento a partir de funções disponibilizadas pela biblioteca da *Microsoft*. É ainda de salientar que o emparelhamento é realizado através da aplicação original da marca, pelo que a sua instalação é necessária.

A partir da Figura 17 pode observar-se que esta aplicação contém apenas três botões, sendo portanto bastante simples. A sua utilização também é bastante acessível: em primeiro lugar é imprescindível pressionar o botão “*Heart Rate Consent*”. Para além do sensor de frequência cardíaca, outros sensores da *Microsoft Band 2* requerem o consentimento do utilizador para que possam adquirir dados. Neste caso apenas é necessário o consentimento para utilizar o sensor de frequência cardíaca, uma vez que o sensor de temperatura corporal não requer consentimento, e apenas são utilizados estes dois sensores na aplicação. Pressionando o botão surge uma janela, tal como na Figura 18, em que o utilizador deverá seleccionar “*Yes*”.

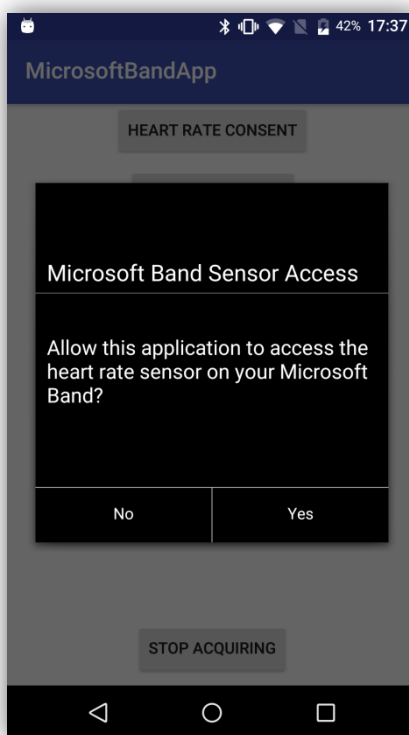


Figura 18 – Pedido de consentimento de acesso ao sensor de frequência cardíaca da *Microsoft Band 2*.

Após o utilizador dar consentimento e pressionar o botão “*Start Measuring*” é iniciada a medição destes parâmetros vitais pela pulseira, cujos valores podem ser vistos na interface da aplicação e são atualizados em tempo real, sendo dado um sinal ao utilizador de que a medição foi iniciada. Este sinal consiste na vibração da pulseira durante um curto período de tempo. A medição destes parâmetros em tempo real é programada em tarefas assíncronas que recorrem à utilização de funções *Microsoft* específicas para a *Microsoft Band 2*.

A Figura 19 é representativa desta interface e apresenta as medições destes parâmetros em tempo real.

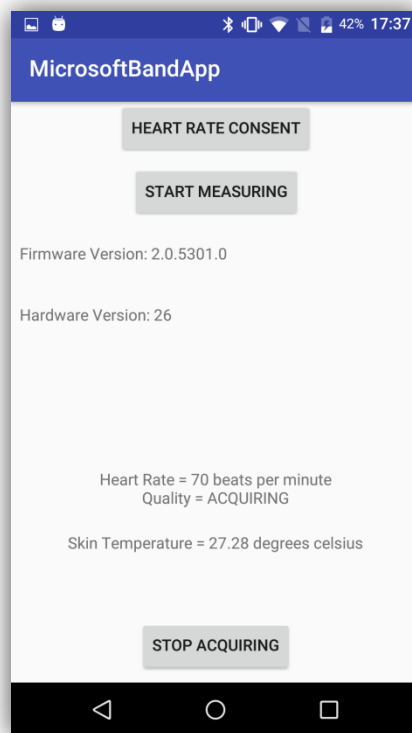


Figura 19 – Indicação em tempo real da frequência cardíaca e da temperatura corporal medidas.

Pode assim verificar-se que o utilizador da pulseira, no momento da aquisição, apresentava uma frequência cardíaca de 70 batimentos por minuto e uma temperatura corporal de 27.28°C. Este valor de temperatura aparentemente baixo deve-se ao facto de a medição ser feita no pulso. Recorde-se da Secção 2.2.1.1 que a temperatura medida à superfície da pele é inferior relativamente à medição feita na axila, o modo mais comum.

A mudança destes parâmetros na interface é conseguida através de um *event listener*, que implementa uma interface para um *change event*, recebendo um *callback* sempre que ocorre uma mudança. No caso da frequência cardíaca, o *BandHeartRateEventListener()* implementa o *onBandHeartRateChanged(BandHeartRateEvent event)*, que por sua vez permite atualizar a interface da aplicação. É ainda de referir que, através das funções *bandClient.getFirmwareVersion()* e *bandClient.getHardwareVersion()* obtêm-se as versões de *firmware* e *hardware* da pulseira, que também podem ser vistas na interface da aplicação.

Como última etapa, para terminar as medições, o utilizador pode pressionar o botão “*Stop Acquiring*”. A pulseira cessará de imediato a aquisição de sinais, e será dada indicação na interface da aplicação de que estes não estão a ser adquiridos, tal como se pode exemplificar na Figura 20.

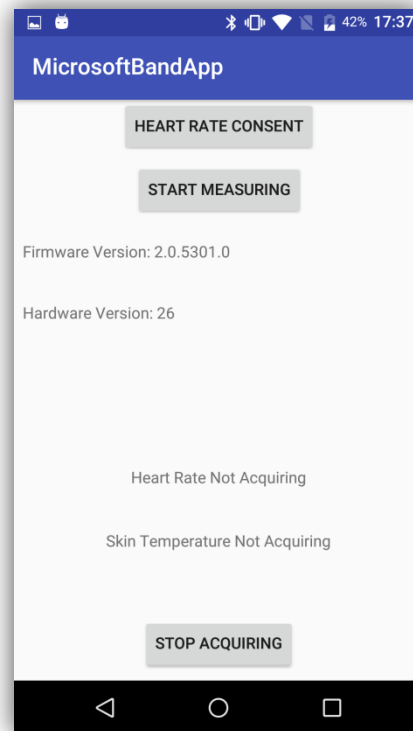


Figura 20 – Interface da aplicação com indicação de que não estão a ser adquiridos sinais pela pulseira.

Um dos objetivos da aplicação seria, ainda, a obtenção em tempo real do RSSI da pulseira emparelhada com o dispositivo móvel que está a ser usado. No entanto, após alguma pesquisa, concluiu-se que a aquisição apenas é possível se os dois equipamentos não se encontrarem emparelhados, ou seja, apenas quando o *smartphone* está em busca de dispositivos *Bluetooth* nas proximidades. Desta forma, não é possível obter o RSSI da pulseira, pelo que este ponto constitui mais um obstáculo de implementação do trabalho.

A solução passou assim por usar uma aplicação já existente, desenvolvida pelo Laboratório de Automática e Sistemas do Instituto Pedro Nunes, abordada de seguida, na Secção 4.4.2.

4.4.2 Integração do algoritmo de localização *indoor* na aplicação *Android* desenvolvida

Após estabelecer comunicação entre a aplicação *Android* desenvolvida e a pulseira da *Microsoft*, e sendo possível a recolha dos parâmetros vitais pretendidos, o grande desafio foi o algoritmo de localização. Nesta etapa foi usada uma aplicação desenvolvida pelo Laboratório de Automática e Sistemas, capaz de localização *outdoor* e *indoor* em alguns edifícios do IPN, bem como capaz de indicar ao utilizador outro tipo de informações menos relevantes, como informações relativas aos vários edifícios e laboratórios existentes.

Esta aplicação, denominada *WiFind*, recorre à utilização de um dispositivo móvel *Android*, de técnicas de localização e de lógica do lado do servidor para a localização no interior de edifícios. O método de localização *indoor* utilizado é o de *fingerprinting* com

recurso à tecnologia *Wi-Fi*, e é usado um *Simple Object Access Protocol* (SOAP) para expor os *webservices* usados para a calibração do sistema e para as rotinas de localização.

Apesar de o método de *fingerprinting* ter já sido abordado no Capítulo 2, na secção seguinte é feita uma abordagem mais alargada, dando-se destaque às fases de treino e utilização, bem como a algumas limitações ainda a ultrapassar.

4.4.2.1 Método de *fingerprinting* aplicado à localização *indoor*

Um dos objetivos chave deste projeto é a localização do doente no serviço de urgência em que se encontra para que, caso se verifique uma degradação significativa no seu estado de saúde, o profissional de saúde seja de imediato avisado e o possa localizar fisicamente no espaço de forma eficiente.

No Capítulo 2 foram analisados vários métodos de localização *indoor*, bem como tecnologias passíveis de serem utilizadas, tendo-se concluído que o método de *fingerprinting* constituía a melhor solução, em conjunto com qualquer uma das tecnologias abordadas, embora a tecnologia *Wi-Fi* tenha sido merecedora de destaque. Apesar de esta não ter sido a razão principal que levou à utilização desta tecnologia, o método de localização *indoor* selecionado é o método de *fingerprinting* com recurso à tecnologia *Wi-Fi*. O que motivou esta escolha foi a utilização deste sistema de localização *indoor* (por *fingerprinting* e através de *Wi-Fi*) desenvolvido pelo LAS, o que será explicado na Secção 4.4.2.2.

Este método, por ter recurso a *Wi-Fi*, tem tido alguma distinção recentemente. Um dos motivos é o facto de, ao contrário dos métodos tradicionais de trilateração e triangulação, não ser exigida linha de vista (He & Chan, 2016), sendo a sua precisão menos afetada pelo movimento de pessoas e pelos efeitos de propagação do sinal. Outro motivo é a larga utilização de equipamentos eletrónicos capazes de comunicação *Wi-Fi*, reduzindo os custos de infraestruturas e aproveitando estes equipamentos como alvos de localização. Pode, assim, verificar-se que este sistema é mais robusto, preciso e barato relativamente a outro tipo de sistemas de localização *indoor* (Luo, Hoerber, & Chen, 2013).

O método de *fingerprinting* requer duas fases, sendo que na primeira fase, *offline*, é feito um planeamento da área onde irá ser utilizado o sistema de localização para criar um treino de *fingerprints* (O. C. Reyes, 2014). Este treino é feito a partir do RSSI dos vários sinais *Wi-Fi* de pontos de acesso estrategicamente colocados no espaço, ocupando assim pontos de referência. Cada um destes sinais representa uma *fingerprint* e é guardado numa base de dados para ser utilizado posteriormente na fase *online*. Durante a fase de treino é necessário ter em atenção a densidade de pontos de treino: não é aconselhável reduzir a densidade de pontos de levantamento do local, ou o levantamento direto em larga escala, embora a fase de treino seja bastante trabalhosa e consuma bastante tempo (He & Chan, 2016). Este tipo de sistemas, para conseguirem uma elevada precisão e resolução, exigem um treino minucioso, embora isso acarrete custos elevados de configuração e manutenção do sistema em questão (Luo et al., 2013).

Na fase *online*, o dispositivo móvel que se encontra na área de localização é capaz de ler o RSSI recolhido pelos vários pontos de acesso e criar assim um vetor média das medidas recolhidas. É de salientar a importância de efetuar a média dos sinais lidos, uma vez que por si só cada um dos sinais não é suficiente para identificar uma *fingerpint*. De seguida, o vetor obtido é comparado com os valores de treino guardados na base de dados e, através de algoritmos de posicionamento, é identificada uma localização aproximada do utilizador (O. C. Reyes, 2014), baseada no vetor da base de dados que mais se assemelha ao vetor medido (He & Chan, 2016).

Na Figura 21 pode verificar-se a existência de três pontos de acesso, representados por *routers*. O sinal emitido de cada um dos *routers* é recebido pelo dispositivo *Wi-Fi* do utilizador sob a forma de um vetor, e enviado para um servidor, responsável por devolver a localização do utilizador através do vetor de treino mais semelhante ao vetor obtido.

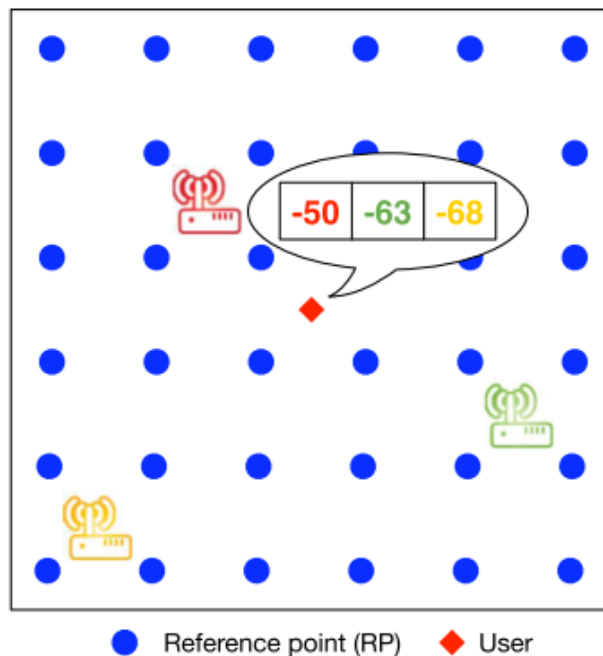


Figura 21 – Sinais obtidos a partir de três pontos de acesso em dBm (Fonte: (He & Chan, 2016)).

Existem alguns fatores condicionantes da medida *online* do sinal *Wi-Fi*, que para um sistema mais preciso podem ser monitorizados de modo a que seja possível atualizar o sinal recolhido de acordo com as condições do ambiente em que o sistema é aplicado. Estes fatores incluem a presença de pessoas no local, os níveis de humidade do ar e o facto de existirem portas abertas ou fechadas. O uso de sensores poderá auxiliar na identificação destas condicionantes (He & Chan, 2016).

Há ainda sistemas que permitem *feedback* humano, ou seja, em que o utilizador do sistema, perante a visualização incorreta da sua localização, pode reportar o erro e até sugerir uma nova localização. Este tipo de aplicações têm a vantagem de adaptação a novas infraestruturas ou a um novo ambiente *wireless*, podendo atualizar a sua base de dados apenas

através da intervenção dos seus utilizadores (Luo et al., 2013). Uma outra preocupação que existe relativamente a este tipo de sistemas de localização *indoor* é a energia consumida pelo *scanning* e transmissão de dados pelos dispositivos *Wi-Fi*. Torna-se, assim, necessário um equilíbrio entre a frequência de busca a pontos de acesso e o número de pontos de acesso colocados na área de aplicação do sistema (He & Chan, 2016).

A aplicação *WiFind* retorna a localização do utilizador de acordo com este método específico e, de seguida, será explicada a sua integração na aplicação *Android* desenvolvida.

4.4.2.2 Integração do algoritmo de localização

Uma vez que para este projeto um dos objetivos é a localização *indoor*, a aplicação *WiFind* integrada na aplicação *Android* para a *Microsoft Band 2* já desenvolvida, e adaptada ao que era pretendido. Reforça-se aqui o facto de que o desenvolvimento de um algoritmo de localização *indoor* não era um dos objetivos deste projeto, mas sim a integração de uma solução de localização *indoor* no sistema desenvolvido.

A aplicação *WiFind* é constituída por várias atividades, classes, serviços e interfaces, e integra um SOAP que permite fazer pedidos de localização ao servidor através dos sinais *Wi-Fi* recebidos pelo dispositivo móvel que se está a utilizar, e que é no fundo um protocolo de comunicação entre diferentes aplicações com recurso a HTTP. As características do protocolo SOAP são abordadas na Secção 4.6. Assim, o primeiro passo consistiu na análise atenta das atividades e classes desta aplicação com o intuito de seleccionar o que realmente era necessário para a localização *indoor*, e qual o funcionamento deste algoritmo. É ainda de referir que este algoritmo de localização utiliza a planta do edifício para que o utilizador possa visualizar a sua localização, pelo que esta é um requisito necessário. No exemplo aqui descrito, usou-se a planta do IPN.

O algoritmo de localização *WiFind* pode ser, assim, descrito em seis passos:

1. Na *MainActivity*, a atividade principal da aplicação, existe um menu com alguns ícones. Ao clicar no ícone da localização *indoor* (Figura 22(a)) surge uma janela com os vários edifícios do IPN, pelo que o utilizador terá de seleccionar o edifício em que se encontra. É iniciada, assim, a atividade *LocationActivity*.
2. Nesta atividade é feito o display do mapa do edifício de acordo com o piso em que o utilizador se encontra. Para a atividade de localização é usado o serviço *LocalService*, que verifica a conexão *Wi-Fi* existente, e que é iniciado após o utilizador clicar no ícone da Figura 22(b).
3. Na interface *InterfaceLocation*, usada por este serviço, é declarada a *Uniform Resource Locator* (URL) do servidor para onde é feito o pedido de retorno da posição do utilizador.
4. Após o pedido, é a classe *NewSoapServices* que faz o *request* e obtém o resultado, que é constituído pela latitude e longitude em que o utilizador se

encontra, entre outros. A localização é enviada novamente para a *LocationActivity*.

5. A partir da classe *CoordinateConversions* são calculadas, em pixéis, as coordenadas de localização do utilizador.
6. É através da função *DisplayMarker* que o ponto é desenhado no mapa da aplicação, demarcando o local onde o utilizador se encontra.

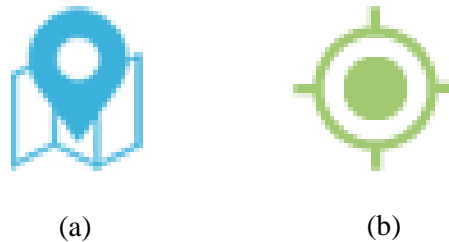


Figura 22 – Ícones de destaque na aplicação *Android* desenvolvida. (a) Início da atividade *indoor*. (b) Permitir localização *indoor*.

A localização do utilizador no interior do edifício é atualizada periodicamente.

A aplicação *WiFind* recorre a bibliotecas e imagens necessárias ao funcionamento do algoritmo, que foram importadas pela aluna para a aplicação da *Microsoft Band 2*. Foi ainda necessário corrigir alguns erros e colmatar a falta de alguns recursos no código.

Após esta explicação acerca do funcionamento da aplicação desenvolvida, o seu modo de utilização é bastante intuitivo. Semelhante à primeira versão da aplicação desenvolvida e exposta na Secção 4.4.1, uma das diferenças relativamente a esta é que à atividade principal (que na primeira versão da aplicação corresponde à interface representada na Figura 17) foi acrescentado um campo denominado de “*Episódio*”, que o utilizador terá de preencher com o ID do episódio correspondente ao doente que pretende monitorizar e que é apresentado na *dashboard* sempre que é criado um novo doente. O motivo da inclusão deste campo é apenas criar uma relação entre os dados que estão a ser recolhidos e enviados para a base de dados de acordo com o *web service* descrito na Secção 4.6 e o episódio em questão. Este campo é de preenchimento obrigatório, pois o funcionamento da aplicação depende do ID indicado, e depois de validado não pode ser modificado, sendo necessário reiniciar a aplicação. Uma outra diferença relativamente à primeira versão da aplicação é a inclusão do menu de localização *indoor*, como se verifica na Figura 23.

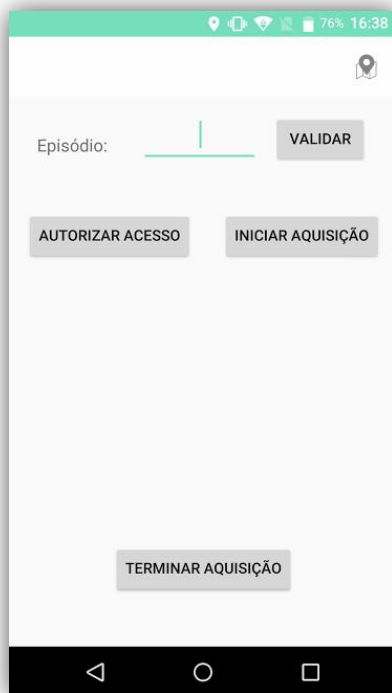


Figura 23 – Atividade principal da aplicação *Android* para a *Microsoft Band 2*.

Tendo sido preenchido o referido campo, validado, e feito o procedimento já conhecido para iniciar a aquisição dos dados, resta apenas proceder como já descrito: clicar no botão do menu de localização *indoor* semelhante ao da Figura 22 (a), seleccionar o edifício em que o utilizador se encontra de entre os edifícios da lista de plantas que surge no ecrã, tal como na Figura 24, seleccionar o piso e clicar no botão da Figura 22 (b) para iniciar o algoritmo de localização.

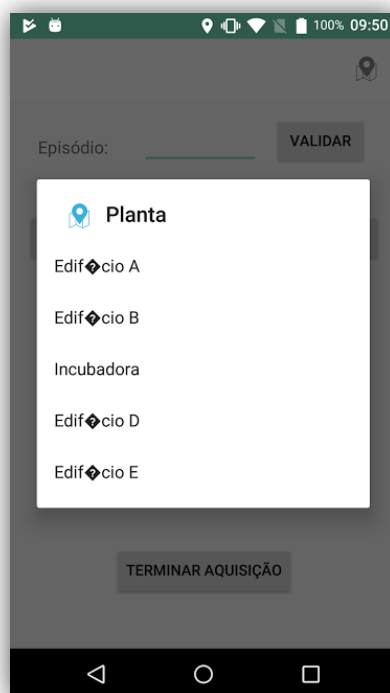


Figura 24 – Lista de edifícios previstos no funcionamento da aplicação *Android* para a *Microsoft Band 2*.

A Figura 25 representa a título de exemplo a planta do Piso 1 do Edifício A do IPN. A localização do utilizador está representada através do ponto azul.

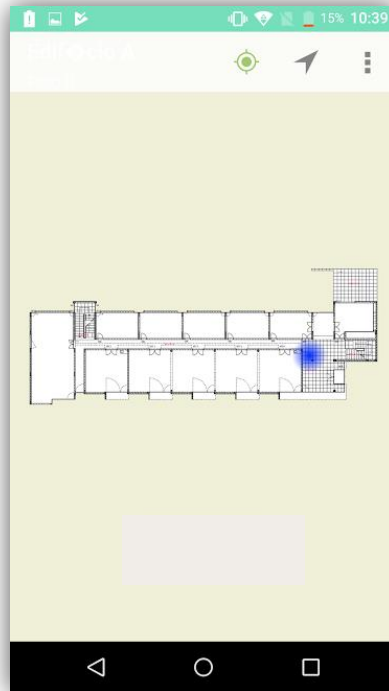


Figura 25 – Planta com indicação da posição do utilizador da aplicação.

Existem ainda alguns aspetos que é necessário ter em atenção. Em primeiro lugar, uma vez na atividade de localização não se deverá regressar à atividade principal. A recolha e envio dos parâmetros vitais do doente funciona de forma assíncrona e, portanto, continua a correr quando há mudança de atividade; o mesmo não acontece com o algoritmo de localização, pelo que se deverá ter em atenção este aspeto para o bom funcionamento da aplicação. Um outro aspeto relevante é que o sistema precisa de ser calibrado, pelo que apenas funciona num edifício para o qual já foram recolhidas medidas de teste, como indica o método de *fingerprinting*. Assim, neste caso, o sistema implementado apenas funciona no Edifício A do IPN, pelo que foi neste edifício que foram realizados todos os testes necessários.

Um outro componente do sistema é a *dashboard* que permite visualizar os dados de todos os doentes que estão a ser monitorizados, pelo que será realizada uma abordagem à sua implementação na Secção 4.5.

4.5 Dashboard

Com o objetivo de guardar os dados do doente, nomeadamente os valores da frequência cardíaca e da temperatura corporal recolhidos pela pulseira, e outras informações relevantes relativas não só ao doente como a outros componentes do sistema, tornou-se necessária a implementação de uma base de dados. Foi então feita uma análise que aborda alguns

conceitos de bases de dados e a realização de modelos necessários à sua implementação, nomeadamente o modelo de entidade-relacionamento e os modelos conceptual e físico de dados.

4.5.1 Conceção da base de dados

A conceção de uma base de dados adequada a uma aplicação particular engloba um planeamento exigente, atendendo às necessidades do projeto. É essencial conhecer a organização da base de dados, as relações que podem existir entre tabelas e as metodologias de planeamento para poderem ser gerados modelos funcionais, simples e facilmente implementáveis.

4.5.1.1 Introdução a bases de dados

Uma base de dados é um conjunto de informação organizada em linhas, colunas e tabelas tendo em vista a facilidade de acesso, gestão e atualização dos dados (Rouse, 2017). A estrutura da informação deve ser tal que torne simples acrescentar e eliminar dados, bem como modificá-los. A extração da informação é feita através de um Sistema de Gestão de Base de Dados (DBMS - *Database Management System*) como resposta a pedidos por parte do utilizador (The Editors of Encyclopædia Britannica, 2005).

Relativamente à forma como uma base de dados deve estar organizada, o uso de tabelas é essencial para evitar a ocorrência de redundâncias. Para entender este conceito importa reter duas definições base: cada linha da tabela corresponde a um registo, e a cada registo pertencem dois ou mais campos, que por sua vez são colunas. Para uma melhor compreensão, é dado um exemplo pelo *site* de ajuda da *Microsoft* (<http://support.office.com>): dada uma tabela em que se pretende armazenar dados de funcionários, cada registo (linha) contém dados de um funcionário diferente e cada campo (coluna) contém informação distinta, por exemplo o nome, endereço, entre outros. Cada funcionário é inserido uma única vez para que haja normalização (Microsoft, n.d.-b). Em linguagem de base de dados, pode ainda dizer-se que cada objeto é uma entidade e a cada entidade correspondem atributos (Suehring & Valade, n.d.). Importa ainda referir que cada campo é definido por um tipo de dados, podendo uma tabela ter campos que diferem entre si no que diz respeito ao tipo de dados. Estes podem, tipicamente, ser do tipo texto ou número, podendo ainda assumir o tipo data/hora ou um formato de contabilidade (Microsoft, n.d.-a).

Dado que uma base de dados pode apresentar mais do que uma tabela, torna-se pertinente saber como é que as tabelas se podem relacionar entre si: embora a natureza dos dados armazenados possa ser distinta, numa determinada base de dados podem existir tabelas com campos em comum (Microsoft, n.d.-a).

4.5.1.2 Relação entre tabelas

De forma a analisar a relação entre tabelas, há que referir que uma entidade é um conjunto de elementos sobre os quais se pretende guardar informação, e que cada uma tem atributos ou campos, que são essa informação (Lopes, 2011). Assim, a relação entre entidades, também conhecida como cardinalidade, pode ser de três tipos:

- **1 para 1 (1:1)** – a uma instância de uma determinada entidade corresponde apenas uma instância de uma segunda entidade (Poolet, 1999). Um exemplo simples é que cada pessoa tem um e só um BI (ou cartão de cidadão), e um BI corresponde apenas a uma pessoa (Lopes, 2011).
- **1 para N (1:N, em que N significa vários)** – cada instância de determinada entidade pode ser relacionada com uma ou mais instâncias de uma segunda entidade (Poolet, 1999). Exemplo desta relação é um produto que pertence apenas a um fornecedor, podendo no entanto esse fornecedor providenciar vários produtos (Chang, 2017).
- **N para M (N:M, em que N e M significam vários)** – várias instâncias de uma entidade podem relacionar-se com várias instâncias de uma segunda entidade. Desta relação pode dar-se como exemplo um pedido de compra de produtos. Este pedido pode incluir vários produtos, assim como o mesmo produto pode ser requisitado por vários pedidos (Poolet, 1999).

O modelo que traduz estas relações é o modelo entidade-relacionamento, em que as entidades são representadas geralmente por polígonos, e o relacionamento por uma linha que os liga entre si (Rouse, 2005). O modelo entidade-relacionamento para o sistema em questão encontra-se na Figura 26.

Um outro conceito que requer uma análise cuidada é o conceito de chaves. Existem chaves primárias e chaves externas, que consistem num ou mais campos, e que fazem parte da relação entre tabelas. Uma chave primária identifica de forma única cada registo armazenado na tabela, sendo tipicamente um ID ou um código. Cada tabela pode ter apenas uma chave primária. Uma chave externa permite estabelecer a correspondência entre uma chave primária e os valores que lhe dizem respeito noutra tabela. Pode ainda ser uma chave composta, isto é, composta por mais do que uma coluna, atendendo a que possam existir características comuns. Cada tabela pode, assim, ter mais do que uma chave externa.

As chaves primárias e externas são peças fundamentais em bases de dados, sendo que são o que permite formar uma correspondência entre dados de tabelas relacionadas. O uso de relações permite evitar dados repetidos o que faz com que não exista ambiguidade, e uma maior facilidade de compreensão da informação (Microsoft, n.d.-a).

Após esta breve introdução a bases de dados importa estudar qual a metodologia a utilizar na implementação das tabelas pretendidas, apresentada na Secção 4.5.1.3.

4.5.1.3 Metodologia

Para construir uma base de dados funcional e adequada à aplicação pretendida é essencial seguir uma determinada metodologia. Esta metodologia consiste num conjunto de passos, sendo eles:

1. Criar um modelo entidade-relacionamento:
 - a) Identificar as entidades, ou seja, os principais objetos de interesse;
 - b) Traçar relacionamentos;
 - c) Associar atributos das diversas entidades através dos relacionamentos;
 - d) Verificar possíveis redundâncias do modelo, analisando relações (1:1).
2. Mapear o modelo entidade-relacionamento para construir tabelas (Thomas M. Connolly, 2004);
3. Identificar os atributos de cada objeto;
4. Determinar chaves primárias como identificadores únicos de cada tabela;
5. Definir valores por omissão, caso necessário, valores que podem ser comuns a vários campos;
6. Delimitar quais as colunas que não podem ter valor nulo, que requerem sempre que seja inserida informação.

É de ter em atenção que não pode existir informação repetida; ter a mesma informação em duas tabelas diferentes pode ser sinónimo de problemas quando se pretende efetuar alguma alteração na informação guardada (Suehring & Valade, n.d.).

Seguindo a metodologia apresentada, foram construídos os modelos necessários à implementação da base de dados do projeto.

4.5.1.4 Modelo entidade-relacionamento

A criação do modelo entidade-relacionamento, já exposto anteriormente, é o primeiro passo a seguir para a criação da base de dados do projeto, que vai armazenar diferentes tipos de informação. Atendendo aos requisitos e à arquitetura do sistema que se pretende implementar, em especial aos *mockups* desenhados (Secção 3.3), podem identificar-se as entidades seguintes:

- **Episódio** – Entidade que reúne cada uma das situações de urgência ocorridas;
- **Doente** – Entidade que corresponde a todos os doentes inseridos no sistema;
- **Cor da pulseira** – Entidade que permite a identificação da cor da pulseira atribuída ao doente e que determina a sua prioridade de acordo com o protocolo de Manchester;
- **Parâmetros** – Entidade que define a frequência cardíaca e a temperatura corporal para as quais, para determinado doente, deverá ser emitido um alerta pelo sistema;
- **Sinais** – Entidade que agrupa os dados recolhidos pelo dispositivo de monitorização de sinais vitais (pulseira);

- **Alarme** – Entidade que armazena todas as situações de alarme ocorridas para cada doente em determinado episódio de urgência.

Definidas as entidades, o passo seguinte será estabelecer os relacionamentos entre cada uma delas. Foi, assim, gerado o modelo entidade-relacionamento da Figura 26. Este modelo foi criado com o auxílio do *software Visual Paradigm* (<https://www.visual-paradigm.com/>).

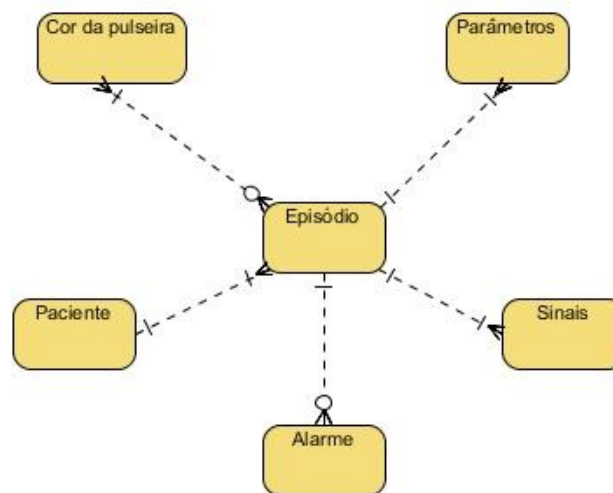


Figura 26 – Modelo entidade-relacionamento correspondente ao sistema a implementar.

Observando o modelo da Figura 26 pode, assim, afirmar-se:

1. Cada doente inserido no sistema com determinado ID pode ter mais do que um episódio de urgência, sendo que cada episódio corresponde a um único doente;
2. Durante um episódio de urgência pode ser atribuída mais do que uma cor de pulseira (se houver degradação da saúde do doente poderá ser necessário alterar a sua prioridade), embora possam existir vários doentes aos quais foi atribuída a mesma cor;
3. Para cada doente são definidos parâmetros de alerta, sendo que este conjunto de valores para cada doente pode ser alterado durante o episódio de urgência.
4. São recolhidas várias medições de sinais para cada doente, que dizem respeito a um único doente e a um único episódio de urgência;
5. Para cada episódio de urgência, associado a um doente, pode haver ocorrência de alarmes, podendo surgir vários alarmes em cada episódio.
6. Conclui-se ainda que as relações entre entidades são de (1:N) ou de (N:M), podendo relacionar-se o doente com todas as outras entidades, direta ou indiretamente através do episódio de urgência que lhe está associado.

Importa ainda salientar que algumas situações previstas durante o planeamento deste modelo não tinham sido previstas ao desenhar os *mockups*.

Em primeiro lugar, a alteração da cor da pulseira durante o episódio de urgência não tinha sido previsto, sendo no entanto importante para que, em caso de degradação da saúde do doente, se possa estabelecer uma nova prioridade e o doente possa ser atendido mais rapidamente. Um outro cenário não previsto foi a alteração dos parâmetros de alerta durante o episódio: imagine-se uma situação em que o doente tem febre, e quando deu entrada o alerta estava definido para uma temperatura corporal de 39 °C. Se houver um alarme para esta temperatura, será necessário ajustar o valor da temperatura para a qual deverá ser dado o próximo alerta; caso contrário o sistema estará a emitir um alerta continuamente até que a temperatura corporal do doente desça.

O próximo passo consiste em associar atributos a cada uma das entidades e em traçar as relações entre elas através de um modelo conceptual de dados, discutida na Secção 4.5.1.5.

4.5.1.5 Modelo Conceptual de Dados

O modelo conceptual de dados envolve a criação de entidades, com os atributos de cada uma delas, e a sua associação com outras entidades, formando assim um mapa de conceitos, relações e restrições (Hutson, 2016). Deverá ser um modelo universal e de fácil compreensão, construído a partir de notações genéricas ou símbolos que permitam ao utilizador uma assimilação intuitiva (Ponniah, 2007). Para a realização deste modelo foi usado o *software Power Designer* (<http://powerdesigner.de/en/>). A escolha deste *software* recaiu no facto de ser uma ferramenta que permite gerar automaticamente o modelo físico de dados a partir do modelo conceptual de dados, para além de conseguir detetar possíveis erros do modelo. O modelo conceptual de dados está representado na Figura 27.

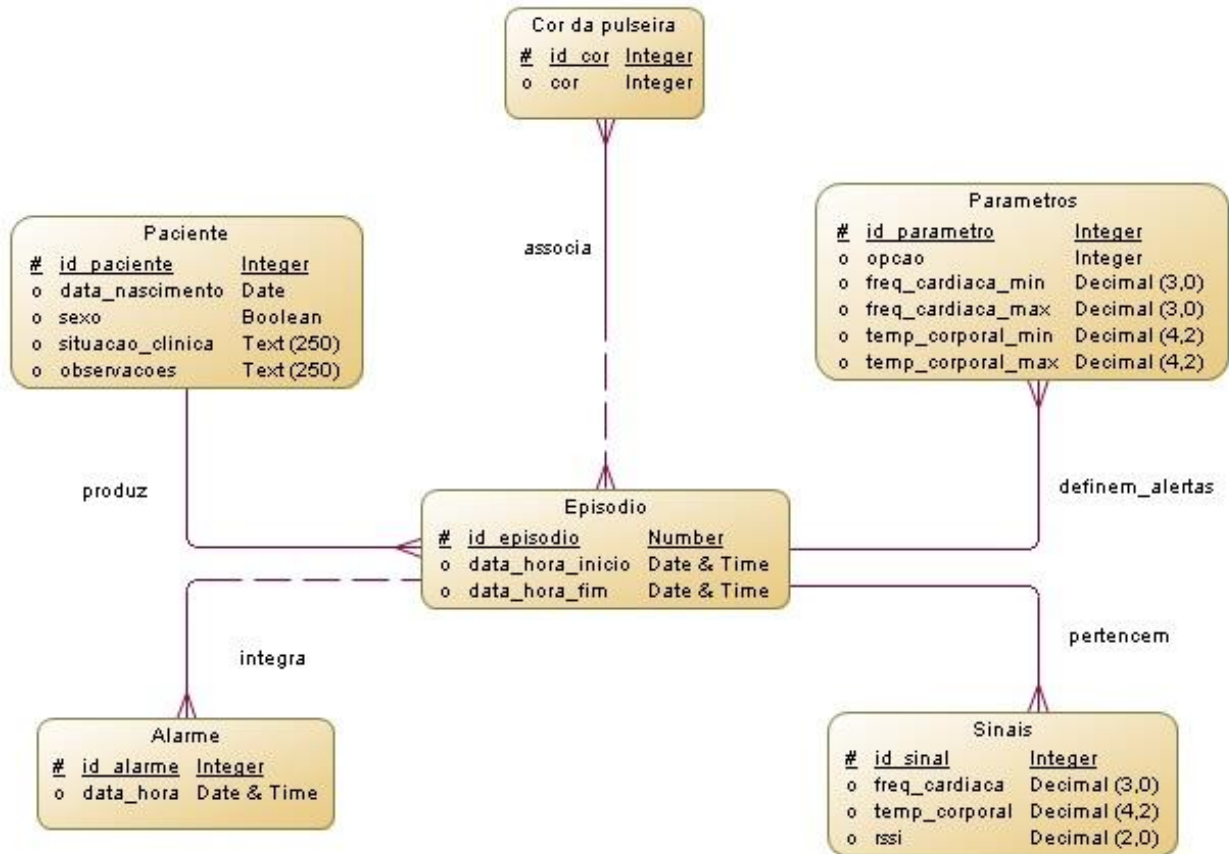


Figura 27 – Modelo conceptual de dados do sistema a implementar.

Pela observação atenta da Figura 27 podemos verificar que o modelo conceptual de dados é de todo semelhante ao modelo entidade-relacionamento; apenas são apresentados todos os atributos de cada entidade, e o tipo de dados de cada um. É de notar que cada uma das entidades é identificada por um ID, que constitui uma chave primária. O tipo de dados é maioritariamente numérico, dado que a maior parte dos dados que se pretendem guardar na base de dados são referentes a sinais como a frequência cardíaca e a temperatura corporal do doente, bem como o RSSI da pulseira. Existem também datas, que marcam o início e o final do episódio de urgência do doente, bem como a ocorrência de alarmes.

Tendo sido realizados já os modelos de entidade-relacionamento e conceptual de dados, segue-se a conceção do modelo físico na Secção 4.5.1.6.

4.5.1.6 Modelo físico de dados

O modelo físico de dados representa a construção do sistema, incluindo detalhes complexos acerca das estruturas de dados, dos quais se destacam a organização de dados e ficheiros e parâmetros de armazenamento de dados. É este modelo que dita as capacidades e funcionalidades de um DBMS (Ponniah, 2007). O modelo físico gerado pelo *software* é o da Figura 28.

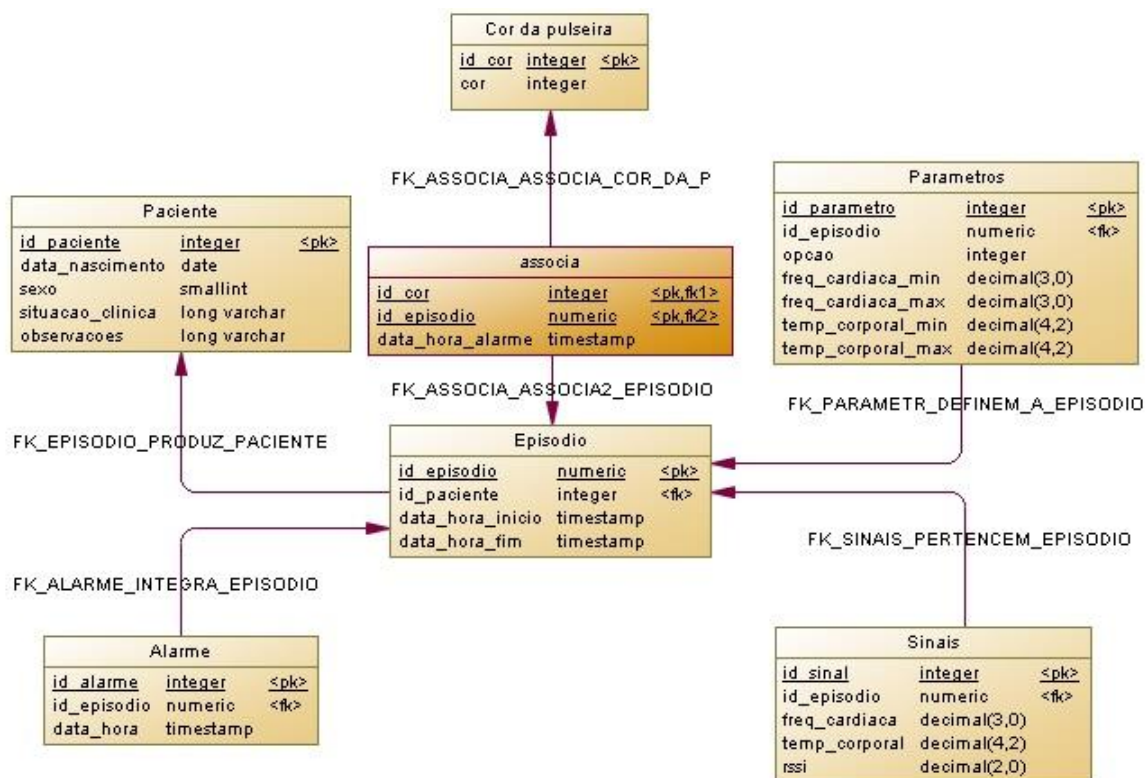


Figura 28 – Modelo físico de dados gerado automaticamente pelo *software* a partir do modelo conceptual.

Após uma análise do modelo, podemos verificar que a diferença entre o modelo físico e o modelo conceptual é a identificação das chaves primárias e externas, representadas por *pk* (de “primary key”) ou *fk* (de “foreign key”), respetivamente. Lembra-se que as chaves são a base do relacionamento, sendo que são o que permite formar uma correspondência entre dados de tabelas relacionadas. Verifica-se, assim, que na tabela de cada entidade existe uma chave externa, que representa os dados de uma segunda entidade que se relacionam com a primeira, e que esta chave externa é a chave primária da entidade à qual pertence.

Foi ainda associada uma variável temporal à tabela que relaciona o episódio com a cor da pulseira atribuída ao doente para que se possa saber em que momento do episódio de urgência a prioridade do doente foi alterada. Esta tabela foi gerada automaticamente pelo *software*, uma vez que a relação era de N:M, ou seja, de vários para vários, sendo que cada episódio pode ter mais do que uma cor de pulseira, e cada uma das cores pode fazer parte de vários episódios em simultâneo.

Após terem sido revistos todos os modelos criados, o último passo é a geração do código da base de dados, que mais uma vez é feita automaticamente pelo *software* utilizado.

4.5.2 Implementação da *dashboard*

Na implementação da *dashboard* a *framework* principal necessária foi o *Laravel*, que por sua vez requer a instalação de outras ferramentas para que possa ser utilizada.

Esta *framework* recorre, em primeiro lugar, ao *Composer* para gerir as suas dependências, pelo que foi indispensável a instalação desta ferramenta, que, em suma, permite declarar as dependências das bibliotecas necessárias, bem como instalar e atualizar os pacotes necessários automaticamente no projeto (“Composer - Introduction,” n.d.).

Para utilizar o *Composer* torna-se necessária a instalação de um *web server*, neste caso o *WampServer*. WAMP pode ser traduzido por “*Windows, Apache, MySQL e PHP*”, sendo o *Windows* o sistema operativo com o qual é usado, *Apache* um servidor que permite o desenvolvimento e teste de páginas *web* sem que estas sejam publicadas na *Internet*, *MySQL* a base de dados e *PHP* a linguagem de programação que permite aceder e interagir com a base de dados, permitindo criar uma página dinâmica sem a publicar (Christensson, 2013). A razão pela qual foi necessária a instalação deste servidor foi o facto de o *Composer* necessitar do *PHP* para funcionar (Biler, 2013), pelo que se não se seguir este passo não será possível utilizar o *Laravel*.

Através da janela de comandos do *Composer* foi possível criar um novo projeto, que pode ser visualizado a partir de um *software* de edição de texto, tendo sido utilizado neste projeto em especial o *Sublime Text*. Pode verificar-se, assim, que inicialmente um projeto é composto por várias pastas e ficheiros, universais para todos os projetos. Posteriormente foi o programador o responsável por criar os restantes ficheiros necessários ao projeto em questão.

O passo seguinte consistiu na criação e configuração da base de dados. Esta foi criada no servidor *phpMyAdmin*, um *software* gratuito e que oferece suporte à maioria das funcionalidades do *MySQL* (“Bringing MySQL to the web,” n.d.). É de referir que a base de dados não foi criada manualmente neste servidor através de linguagem *SQL*, uma vez que o *software* usado no desenvolvimento do modelo físico de dados permitia gerar o código da base de dados automaticamente; apenas houve necessidade de alguns ajustes mais tarde durante a implementação para que alguns problemas pudessem ser ultrapassados. A base de dados pode ser configurada no ficheiro *.env*.

Uma outra configuração importante foi a que diz respeito à *timezone*, que por omissão está definida como *Coordinated Universal Time* (UTC), que em horário de verão representa menos uma hora em relação à *timezone* do Meridiano de Greenwich (que engloba Portugal). Esta configuração foi feita no ficheiro *config/app.php*, e mostrou-se relevante na interação com a base de dados.

Após as instalações e configurações iniciais, procedeu-se à implementação da *dashboard*, que será enunciada de seguida, respeitando a ordem das etapas necessárias.

4.5.2.1 Desenvolvimento das interfaces

Para a implementação da *dashboard*, o primeiro passo foi o desenho das interfaces, de acordo com os *mockups* desenhados no Capítulo 3. Para este passo foi criada em */resources/views* uma pasta denominada *interfaces*, e dentro desta pasta um ficheiro do tipo *.blade.php* para cada uma das interfaces, sendo este desenvolvido em linguagem *HTML*. Neste ficheiro, denominado de *blade template*, foram definidos todos os elementos da interface em questão:

desde botões a *labels*, tabelas, imagens, campos de escrita de texto, entre outros. Foi apenas necessário indicar qual a natureza do objeto, e dar-lhe uma classe, um ID e um nome para que pudesse ser identificado e referenciado noutra *script* do projeto. A sintaxe utilizada foi comum a todos os objetos, e esteve de acordo com o exemplo do campo de inserção de dados presente na Figura 29.

```
<input class="box" type="date" name="data_nascimento" id="bdaydate" value="{{ old('data_nascimento')}}">
```

Figura 29 – Exemplo de definição de um objeto através da linguagem HTML.

Neste exemplo o objeto é do tipo *input*, com dados do tipo “*data*” e com a classe, o nome e o ID indicados. Foi necessário ainda, em alguns casos, indicar o tipo de dados, o valor do objeto, a sua visibilidade, se está ou não ativo e se é apenas de leitura. O nome do objeto, no caso de campos em que se pretenda interagir com a base de dados, deverá ser o nome da coluna com que se pretende interatuar. A ordem em que os objetos são referenciados é importante, sendo que a interface tem de ser construída a partir do topo e da margem esquerda. É de referir ainda que neste *script* não é feito qualquer tipo de personalização de *design*. No entanto, pode recorrer-se ao *Bootstrap* para uma personalização primitiva, uma vez que esta *framework* inclui já classes que podem ser atribuídas a um objeto, e que têm determinado aspeto predefinido. O *Bootstrap* foi, assim, utilizado nesse sentido.

A personalização foi feita no ficheiro *style.css*, no qual todos os objetos de todas as interfaces são referenciados tipicamente pela classe, ou pelo seu ID, sendo que vários objetos podem pertencer à mesma classe se se desejar que tenham um aspeto semelhante. No CSS são definidas características comuns como margens, cores, sombras, o tamanho dos objetos, o tipo de letra a usar e o alinhamento. Existem muitas mais características que se podem definir de acordo com o que se pretende.

No final, o resultado obtido foram as interfaces com *design* de acordo com o que era pretendido, não havendo no entanto qualquer possibilidade de interação, uma vez que ainda não tinham sido implementadas a ligação à base de dados nem as ações responsivas. Relativamente à *dashboard* implementada, verifica-se que existem algumas alterações no *design* relativamente aos *mockups* desenhados e apresentados na Secção 3.3. Da Figura 30 à Figura 33 estão representadas as diversas interfaces da *dashboard*.



Figura 30 - Interface inicial da *dashboard*.

Figura 31- Interface para inserir um novo doente no sistema.

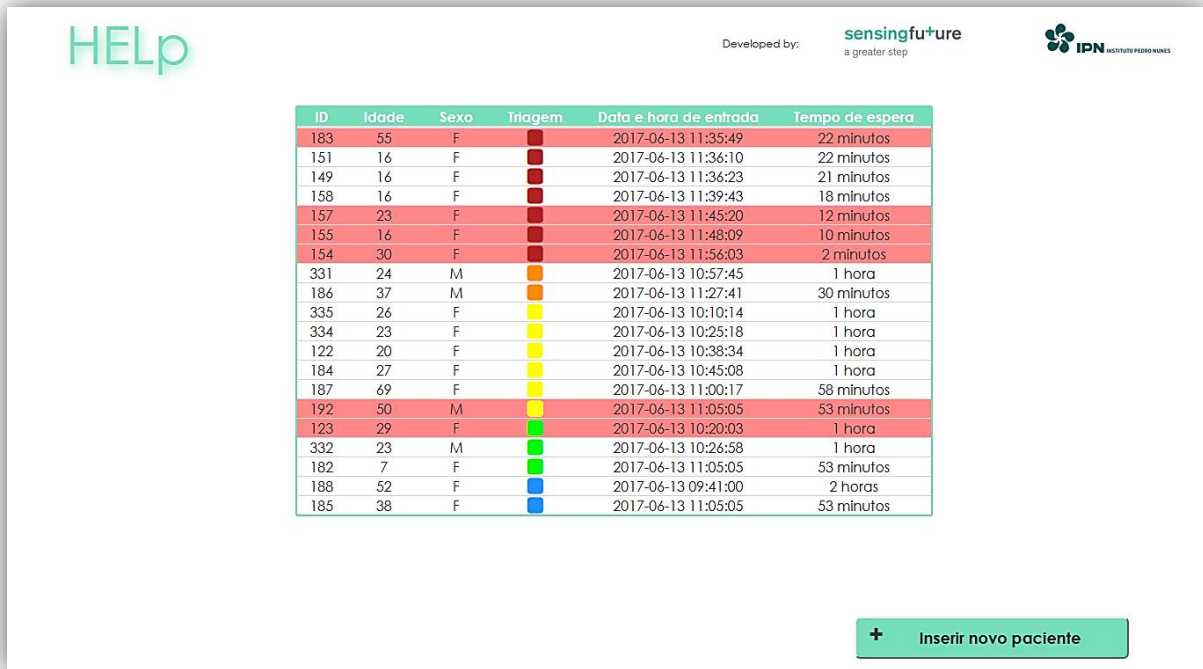


Figura 32 - Lista de doentes do sistema, organizados de acordo com a classificação de triagem e com indicação de alertas.

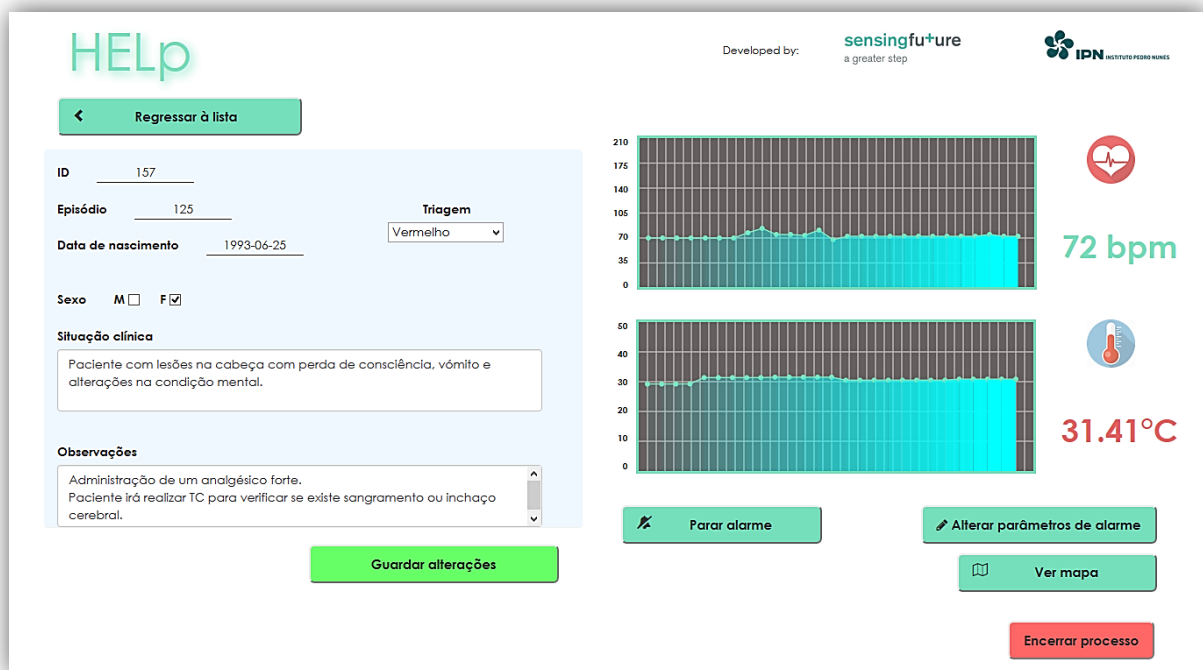


Figura 33 - Informações de um dos doentes da lista, com indicação de alarme na temperatura corporal.

As interfaces reais implementadas, como se pode verificar, são bastante semelhantes aos *mockups* desenhados, como se verifica na interface da Figura 30, que praticamente não sofreu alterações. No entanto, existem algumas diferenças facilmente notórias, referidas de seguida.

Em primeiro lugar, foi acrescentado em todas as interfaces o logotipo do IPN, uma vez que o LAS teve um papel determinante na elaboração deste projeto, como foi referido anteriormente.

Em segundo lugar, comparando a Figura 31 com o *mockup* que lhe deu origem, é de ter em atenção que não é pedida a idade do doente, mas sim a sua data de nascimento, embora não seja uma alteração relevante. No entanto, no que diz respeito à temperatura corporal, não são pedidos limites superior e inferior mas sim uma variação de temperatura. Assim, o profissional de saúde é alertado sempre que a temperatura corporal do doente subiu ou desceu, e não quando atingiu um máximo ou mínimo previamente definidos. Esta alteração foi sugerida pela empresa *Glintt* e tida em conta, dada a experiência desta empresa na área médica e o seu conhecimento relativamente ao funcionamento dos serviços de urgência.

Na lista de doentes a alteração mais notória, de acordo com a Figura 32, é que não existe uma coluna para o estado de alarme: toda a linha é colocada a piscar de forma intermitente sempre que o doente sofre uma alteração do seu estado de saúde.

Por fim, a interface que permite ver as informações relativas a cada doente individualmente, representada na Figura 33, foi a que mais alterações sofreu. Destaca-se em primeiro lugar o botão “*Guardar alterações*”, que permite atualizar a situação clínica do doente, bem como as observações.

É ainda possível alterar a triagem do doente sempre que necessário, pelo que o botão de seleção aparece novamente nesta interface. É dado ainda destaque aos parâmetros vitais do doente, sendo os gráficos de maiores dimensões, e uma *label* indica qual o último valor medido para cada um dos parâmetros, indicando ainda se existe um alarme, ficando com cor vermelha e intermitente aquando da ocorrência. Refere-se ainda que relativamente aos valores da temperatura corporal existia um erro proveniente da pulseira, que fazia com que o primeiro valor recolhido não correspondesse à realidade, sendo bastante inferior. Assim, para prevenir este erro, o primeiro valor de temperatura corporal recebido é descartado, não sendo usado para o cálculo do alarme.

Para parar o alarme é necessário alterar os parâmetros de alarme previamente definidos, o que se pode fazer através do botão “*Alterar parâmetros de alarme*”. Para terminar, o mapa da sala não é imediatamente visível por falta de espaço na interface. A solução encontrada foi colocar um botão que, quando pressionado, abre uma janela de grande dimensão onde é mostrado o mapa e a localização do doente. Destaca-se ainda a colocação, cor e tamanho dos botões de todas as interfaces, que pretendem que a interface seja o mais intuitiva possível e rápida de utilizar.

Dado que, neste ponto, não existia qualquer interação por parte dos objetos da *dashboard*, o passo seguinte seria, assim, a interação com a base de dados, sendo necessário criar essencialmente *migrations* e *seeds*, *models*, *controllers* e *routes*.

4.5.2.2 Interação entre as interfaces e a base de dados

Na interação entre as interfaces desenvolvidas e a base de dados implementada foi necessário aceder às várias tabelas para criar linhas, atualizar colunas, ou simplesmente mostrar alguns dados. Foram, assim, exigidos alguns passos, e a criação de alguns elementos, resumidos de seguida.

4.5.2.3 Migrations e seeds

No projeto em questão não foi necessária a criação de *migrations* e *seeds*. No entanto, é importante reter estes conceitos para projetos futuros. Assim, de uma forma resumida, *migrations* são o controlo da base de dados, que permite estabelecer as colunas de cada tabela e o tipo de dados, entre outros (“Creating a Basic ToDo Application in *Laravel 5 – Part 1*,” 2015). Por sua vez, as *seeds* constituem dados de teste (*Laravel*, n.d.-b), sendo no fundo linhas da tabela em questão com valores padrão para todas as colunas. Como exemplo é apresentado um projeto com duas tabelas, denominadas *projects* e *tasks*, em que cada projeto pode ter mais do que uma tarefa, e cada tarefa pertence apenas a um projeto. Após serem estabelecidas as *migrations* podem visualizar-se as duas tabelas criadas, bem como os campos e tipo de dados que as constituem através da linha de comandos, como representado na Figura 34. É possível ainda visualizar as chaves primárias de cada tabela (representadas por PRI) e qual a correspondência entre tabelas.

```

Projects
+-----+-----+-----+-----+
| Field      | Type                | Null | Key |
+-----+-----+-----+-----+
| id         | int(10) unsigned   | NO   | PRI |
| name       | varchar(255)       | NO   |     |
| slug       | varchar(255)       | NO   |     |
| created_at | timestamp          | NO   |     |
| updated_at | timestamp          | NO   |     |
+-----+-----+-----+-----+

Tasks
+-----+-----+-----+-----+
| Field      | Type                | Null | Key |
+-----+-----+-----+-----+
| id         | int(10) unsigned   | NO   | PRI |
| project_id | int(10) unsigned   | NO   | MUL |
| name       | varchar(255)       | NO   |     |
| slug       | varchar(255)       | NO   |     |
| completed  | tinyint(1)         | NO   |     |
| description | text                | NO   |     |
| created_at | timestamp          | NO   |     |
| updated_at | timestamp          | NO   |     |
+-----+-----+-----+-----+
    
```

Figura 34 – Tabela com as *migrations* implementadas (Fonte: (“Creating a Basic ToDo Application in *Laravel 5 – Part 1*,” 2015)).

Por sua vez, após a definição das *seeds*, é possível ver as duas tabelas com os campos referenciados na Figura 34 mas com linhas preenchidas, ou seja, com valores de teste para projetos e tarefas. A Figura 35 representa a tabela *projects* após a definição de *seeds*.

```
mysql> select * from projects;
```

id	name	slug	created_at	updated_at
1	Project 1	project-1	2015-02-05 01:25:43	2015-02-05 01:25:43
2	Project 2	project-2	2015-02-05 01:25:43	2015-02-05 01:25:43
3	Project 3	project-3	2015-02-05 01:25:43	2015-02-05 01:25:43

Figura 35 - Tabela *projects* após a definição de *seeds* (Fonte: (“Creating a Basic ToDo Application in Laravel 5 – Part 1,” 2015)).

Neste caso foram inseridos na base de dados três projetos, com ID 1, 2 e 3 e com data de criação e atualização.

Como já referido, neste projeto não foi necessário realizar este passo, uma vez que a base de dados já tinha sido gerada através do *software Power Designer*. Também não se utilizaram dados de teste, pois pretendia-se inserir dados na base de dados através da interface que permite inserir um novo doente no sistema. No entanto, as colunas *created_at* e *updated_at* não existiam na base de dados original do projeto, sendo necessárias ao funcionamento da base de dados, e como tal tendo sido criadas posteriormente de forma manual para cada uma das tabelas.

4.5.2.4 Models

Após a definição da base de dados foi necessária a criação de *models*. Para este caso é usado o *Eloquent ORM*, ou seja, a implementação *Object-Relational Mapping* (ORM) incorporada no *Laravel*, que recorre a algumas convenções para que o programador possa ter um *feedback* de dados rapidamente, e comunicar com a base de dados de forma simples (Bautista, 2012). Os *models* são, assim, usados para interagir com a base de dados de forma a guardar, ler, atualizar e eliminar dados de uma tabela, tendo de existir um *model* para cada tabela da base de dados (Chirchir, n.d.). Foram, assim, criados sete *models*, um para cada tabela, sendo em cada um deles referenciada a tabela à qual pertencem, bem como todos os campos desta. É ainda indicada a relação entre a tabela em questão e as restantes, tendo em atenção o modelo físico de dados desenvolvido, referenciando o nome da tabela e qual a chave que permite a relação.

A título de exemplo, é apresentado na Figura 36 o *model* para a tabela “doente”.

```

|<?php
namespace App;

use Illuminate\Database\Eloquent\Model;
use Carbon\Carbon;

class Patient extends Model
{
    protected $table = 'paciente';
    protected $fillable = ['ID_PACIENTE', 'data_nascimento', 'sexo', 'situacao_clinica', 'observacoes'];

    protected $casts = [
        'DATA_NASCIMENTO' => 'date',
    ];

    public function episodio()
    {
        return $this->hasMany('App\Episode', 'ID_PACIENTE', 'ID_PACIENTE');
    }
}

```

Figura 36 – *Model* para a tabela “doente”, com indicação dos campos da tabela e da relação com a tabela “episodio”.

A partir da Figura 36 é possível verificar a referência ao *Eloquent* para a implementação de um *Model*. É ainda referenciada a tabela “doente”, e todos os seus campos, de acordo com o modelo físico de dados. Esta tabela relaciona-se com a tabela “episodio”, implementada no *model* “Episode”, através da chave “id_doente”, sendo que um doente pode ter vários episódios (como indicado por “hasMany”).

Estando implementados todos os *models*, o passo seguinte consistiu na criação de *controllers*.

4.5.2.5 *Controllers*

Os *controllers* são a “central lógica” da aplicação (Dias, 2014), ou seja, são o ponto de ligação entre as *views*, que apenas contêm a interface que se pretende mostrar ao utilizador, e os *models*, que contêm a lógica da aplicação (Brown, 2013). Os *RESTful Controllers* (REST - *Representational State Transfer*) permitem executar as funções básicas ou métodos de uma aplicação, sendo eles criar, extrair, atualizar e eliminar informação (Brown, 2013).

De uma forma geral, foi a partir dos *controllers* que foi inserida informação na base de dados através da criação de uma nova linha, atualizada determinada coluna, obtida informação para ser apresentada numa interface ou simplesmente eliminada informação. No projeto em questão foram implementados cerca de doze *controllers*, sendo os principais o *CreateController*, que gere o processo de introdução de um novo doente na base de dados recolhendo a informação inserida pelo utilizador, o *ListController*, responsável pelo *display* da lista de doentes do sistema a partir de informação da base de dados, e o *PatientController*, que permite mostrar e atualizar alguns dados de determinado doente individualmente.

Alguns dos *controllers* implementados permitem ainda controlar determinados elementos do *design*, como é o caso das *labels* que contém o último valor da frequência cardíaca e da temperatura corporal do doente, e que mudam de cor e ficam intermitentes em caso de alarme. Neste caso, existe um *controller* com a função de, constantemente, perguntar à base de dados qual o último valor medido e enviá-lo para uma função do *Ajax* para que a *label* possa ser atualizada através de código do *JavaScript*. Da mesma forma, outro *controller* é responsável por retirar da base de dados, não só o último valor medido, mas também os parâmetros de alarme definidos inicialmente para cada doente, e realizar a comparação entre eles para verificar se existe uma alteração do estado de saúde de cada um dos doentes.

Para aceder aos *controllers*, bem como para “navegar” entre as várias interfaces, é necessária a criação de *routes*.

4.5.2.6 *Routes*

As *routes*, definidas na diretoria *routes/web.php*, são respostas a um verbo HTTP, sendo que todos os pedidos contêm um verbo que pode ser *get*, *post*, *put*, *delete* ou *any*. À semelhança dos *controllers*, este tipo de *routes* são chamadas de *RESTful Routing*, em que uma possibilidade de uso é o envio de pedidos de URL com ou sem informação adicional usando *get* ou *post* (Rees, 2014) (Brown, 2013a). Na Figura 37 é dado um exemplo de uma *route* deste projeto.

```
Route::resource('episode', 'CreateController');
```

Figura 37 – *Route* para o *RESTful Controller CreateController*, tendo como recurso o *model* “*Episode*”.

Neste exemplo apenas é definido um recurso, o *model* “*episode*”, e o *Laravel* automaticamente direciona a aplicação para o *RESTful Controller* a utilizar, neste caso o *CreateController*.

Em determinadas situações, as *routes* necessitam de parâmetros, como é o caso da URL */patient/194*, em que se pretende aceder às informações de um doente em particular, neste caso o doente cujo *id* é 194. A configuração é semelhante à da Figura 37, sendo que o recurso não é “*episode*” mas sim a URL */patient/{id}*, em que *id* é o parâmetro e que corresponde neste caso ao ID do doente. Para além da definição de *routes* em *web.php*, é possível ainda definir *routes* diretamente num *blade template* usando a função *route()* ou usar o método *Form::open()* para aceder a *routes* às quais foi atribuído um nome (Egan, 2016).

No caso deste projeto, todos estes exemplos de *routes* foram empregados para efetuar a comunicação com a base de dados, utilizando os *models* como recurso e os *controllers* para realizar as ações necessárias.

Após esta abordagem a *dashboard* ficou funcional, sendo capaz de salvar, atualizar e obter dados da base de dados do projeto, bem como “navegar” entre todas as interfaces. No

entanto, existe outro aspeto relevante na construção desta plataforma, tendo como objetivo torná-la mais responsiva. Neste sentido foram usadas as *frameworks JavaScript* e *Ajax*.

4.5.3 Ações responsivas

A linguagem *JavaScript* é a linguagem que confere dinâmica das páginas *web*, permitindo o controlo de aplicações terceiras, como é o caso do HTML (Caelum). De facto, esta biblioteca pode ser usada para diversas ações responsivas, permitindo criar gráficos ou apresentar janelas *pop-up* a partir do clique em determinado botão da interface, ações bastante usadas na implementação da *dashboard*. A partir do *jQuery* é possível ainda referenciar objetos da interface através do nome ou ID que lhes foi atribuído no *script* HTML que implementa cada uma das interfaces.

Uma outra ferramenta bastante útil é o *Ajax*, que é usado para fazer um *refresh* da lista de doentes do sistema automaticamente e sem necessidade de atualizar toda a página, ou ainda para detetar a mudança no estado de objetos e a partir daí efetuar uma ação. Este último exemplo tem aplicação na interface da informação do doente, em que o profissional de saúde pode alterar a triagem do doente perante uma situação de degradação do estado de saúde, selecionando uma nova cor de entre as cores existentes. Dado que a aplicação é responsiva, é detetada a alteração da cor e gerado um *pop-up* em resposta, questionando o utilizador para que a mudança seja validada.

O *Ajax* é, ainda, bastante útil por permitir o uso de um *controller*, chamado através da URL que lhe está associada, para questionar a base de dados e obter respostas com a informação necessária, e assim atualizar objetos como *labels* ou gráficos.

Referente ao último exemplo, é apresentada a função *updateLabel*, que é responsável por atualizar as *labels* da frequência cardíaca e da temperatura corporal da interface da Figura 38.

```
function updateLabel(){
var id_episodio = $('[name="id_episodio"]').attr("value");
$.ajax({
    type: 'GET',
    url: '/HElp/public/label/'+id_episodio, //resource
    beforeSend: function(xhr) {
        xhr.setRequestHeader('X-CSRF-TOKEN', $("#token").attr('content'));
    },
    success: function($data) {
        console.log('sucesso');
        var result = $data;//dados que vem do ajax;
        //alert(result);
        var idx = $data.indexOf(',');

        $('#hr').text($data.slice(0, idx)+" bpm");
        $('#st').text($data.slice(idx + 1)+"°C");
    }
});
}
window.setInterval("updateLabel()", 5000);
```

Figura 38 – Função em *JavaScript* com recurso ao *Ajax*.

Assim, de acordo com a Figura 38, o que a função *Ajax* faz é enviar um pedido do tipo GET para a URL indicada, cujo ID do episódio se refere ao episódio do doente em questão. Esta URL é atualizada através de um *controller*, responsável por percorrer todos os valores de frequência cardíaca e temperatura corporal para extrair o último valor medido para cada um dos parâmetros, e retornar um *array* constituído por estes dois valores. Se a função for executada com sucesso, recebe esse *array* na variável *\$data*, que para ser representado tem de ser repartido. A função é executada a cada cinco segundos e no final o valor de cada uma das *labels* é atualizado.

4.6 Comunicação entre a aplicação *Android* e a base de dados

O grande desafio após a *dashboard* se encontrar funcional foi poder visualizar os parâmetros vitais do doente, bem como a sua localização, recolhidos pela aplicação *Android* desenvolvida. Assim, o objetivo residiu em guardar na base de dados os valores recolhidos, associando-os ao episódio de urgência em questão.

Relativamente aos sinais vitais, a solução encontrada foi usar uma REST API para o envio de dados para a base de dados. Por sua vez, em relação à localização, usar-se-ia o *web service* implementado pelo LAS, que como referido continha um SOAP, por uma questão de facilidade e rapidez.

Importa então entender os conceitos de REST API (Secção 4.6.1) e SOAP (Secção 4.6.2), bem como qual a diferença entre eles (Secção 4.6.3).

4.6.1 REST API

Tal como os *controllers* e as *routes* mencionados, REST API é uma aplicação que recorre a pedidos HTTP através dos verbos *get*, *put*, *post* e *delete* (Rouse, 2016). Desta forma, um *RESTful web service* é um conjunto de protocolos que permite a troca de informações entre aplicações ou sistemas distintos e que podem ser escritos em diferentes linguagens de programação (“RESTful Web Services - Introduction,” n.d.).

Atualmente existem inúmeros serviços que recorrem a *RESTful web services*, pelo que este modelo é visto como predominante relativamente aos métodos que utilizam SOAP e *Web Service Definition Language* (WSDL) (Rodriguez, 2015). Alguns dos *sites* que utilizam este tipo de *web services* para interagir com o serviço da *cloud*, de acordo com (Rouse, 2016), são os sites *Amazon*, *Google*, *LinkedIn* e *Twitter*, bastante conhecidos a nível internacional.

Os pedidos realizados pela API são tipicamente em formato XML, podendo ainda os dados ser submetidos em formato JSON se for conveniente (“Getting started with the REST API,” n.d.). No entanto, para a maioria dos casos, apenas é necessário uma URL.

Para testar uma REST API existem várias ferramentas (Mueller, 2013), de entre as quais o *Postman*, um *plugin* do *Google Chrome*, *Karate DSL*, *HttpMaster Express*, *Rest Console*, entre outras, que se destacam pela facilidade com que é possível fazer um teste sem necessidade de aprender uma nova linguagem de programação (Colantonio, 2017).

4.6.2 SOAP

Tal como uma REST API, SOAP é um protocolo de comunicação entre diferentes aplicações a correr em dispositivos distintos e com linguagens de programação distintas que se baseia em HTTP. Consiste, assim, num ficheiro XML, do qual fazem parte um envelope, elemento que o identifica como uma mensagem XML, um cabeçalho, o corpo da mensagem com o *endpoint* e a informação do pedido e um indicador de falha (“XML Soap,” n.d.). O objetivo é encapsular o pedido numa função e criar um processo capaz de receber respostas por parte do servidor. A resposta recebida vem no mesmo formato que o pedido enviado (Quaine, n.d.). De uma forma geral, o funcionamento do SOAP é o da Figura 39. O utilizador faz um pedido SOAP, que utiliza um dos métodos implementado pelo *web service* para obter dados da base de dados do servidor, e assim retribuir o pedido com a resposta obtida.

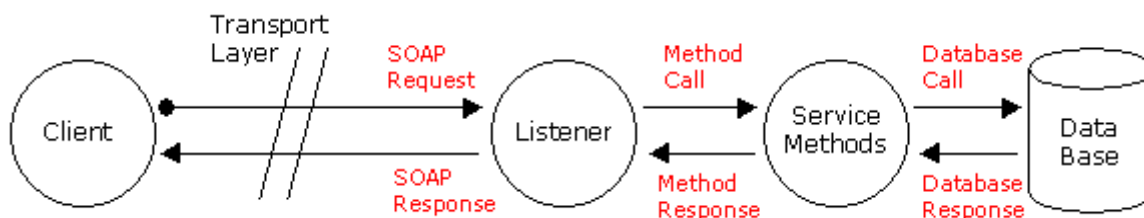


Figura 39 – Funcionamento do SOAP (Fonte: (Quaine, n.d.)).

Os métodos implementados podem ser testados a partir de um ficheiro WSDL, um documento que descreve um *web service*, indicando não só a disponibilidade dos serviços mas também quais os parâmetros requeridos para cada um deles, facilitando assim a utilização do SOAP (Noabeb, 2010). Deste documento fazem parte simples definições, sob a forma de `<definitions targetNamespace="wifind" name="Operations">` (exemplo retirado do ficheiro WSDL da aplicação *WiFind* desenvolvida pelo LAS), e outros elementos relevantes: *types*, que providenciam a definição do tipo de dados usados na troca de mensagens; *message*, a definição dos dados que são transmitidos; *portType* engloba operações de *input* e *output* de mensagens; *binding* a definição do tipo dados das mensagens de determinado *portType*; *port* define um *endpoint* para um *binding*; e por fim *service* agrega todas as portas (Christensen, Curbera, Meredith, & Weerawarana, 2001).

Podemos concluir, a partir desta análise, que REST e SOAP apresentam em comum o facto de recaírem sobre o protocolo HTTP, servindo o mesmo propósito; no entanto, o padrão de mensagens do SOAP é bastante mais rígido. Existem ainda outras diferenças entre os dois protocolos de comunicação (Mueller, 2013), analisadas de seguida.

4.6.3 Diferenças entre REST e SOAP

Após uma pesquisa sobre *web services* verificou-se que existem vários artigos em *sites* e fóruns cuja grande questão é “REST ou SOAP: qual o melhor?”. Na realidade existem diferenças significativas entre estes dois protocolos, pelo que é feita uma breve análise às vantagens de cada um, presente na Tabela 9.

Tabela 9 – Vantagens dos protocolos de comunicação do tipo REST e do tipo SOAP.

Protocolo	Vantagens
REST	<ul style="list-style-type: none"> • Usa os métodos do HTML, pelo que não requer um processo de aprendizagem; • Permite JSON para além da linguagem XML (Brown, 2013c); • Rápido e eficiente, permite formato de mensagens pequeno (Mueller, 2013); • Permite uma boa infraestrutura de armazenamento em cache, melhorando o serviço se as informações guardadas não forem dinâmicas; • É bastante útil para dispositivos como <i>smartphones</i> (Dhingra, 2016).
SOAP	<ul style="list-style-type: none"> • Tem prevista uma lógica de sucesso, que permite ao utilizador lidar melhor com os erros de comunicação (“XML Soap,” n.d.); • É independente de linguagens e plataformas; • Apresenta características <i>standard</i> (Mueller, 2013); • Útil para processos assíncronos (Dhingra, 2016).

Em suma, estes dois tipos de *web services* apresentam vantagens relevantes e a considerar quando se pretende uma aplicação dinâmica, embora REST seja mais simples de implementar em todos os aspetos. Cabe, assim, ao programador decidir qual dos métodos se adapta melhor à aplicação pretendida.

4.5.4 Comunicação entre a aplicação *Android* e a base de dados

No âmbito deste projeto, as duas soluções foram exploradas, em primeiro lugar para o envio dos dados recolhidos pelo dispositivo de medição de sinais vitais para a base de dados do sistema, e em segundo lugar para obter a localização do utilizador.

São, assim, mostrados os passos que foram realizados pela aluna para atingir este objetivo, inclusive algumas dificuldades encontradas.

4.6.3.1 Parâmetros vitais

Com o objetivo de guardar na base de dados do sistema os valores recolhidos pela pulseira dos parâmetros vitais do doente, foi implementada uma REST API. Esta implementação foi

bastante simples recorrendo ao *Postman*, um *plugin* do *Google Chrome*. Através desta ferramenta foi possível preencher uma linha da tabela de forma fácil e rápida.

Em primeiro lugar, foi preciso definir uma *route* no *Laravel* para onde os dados serão enviados, e que, associada a um *controller*, permita guardar os dados na tabela respetiva através da função *store()*. De seguida, no *Postman*, foi indicada a URL para a qual os dados devem ser enviados, e que neste caso teria de incluir o endereço IP correspondente ao servidor e o *model* definido. Por fim, bastou indicar as variáveis que fazem parte do corpo da mensagem e, para efeitos de teste, os valores que se pretendiam guardar. É ainda importante referir que os pedidos realizados são pedidos HTML, tendo de se usar neste caso um POST.

A Figura 40 representa a interface do *Postman* e um exemplo de um pedido ao servidor, bem como a resposta devolvida.

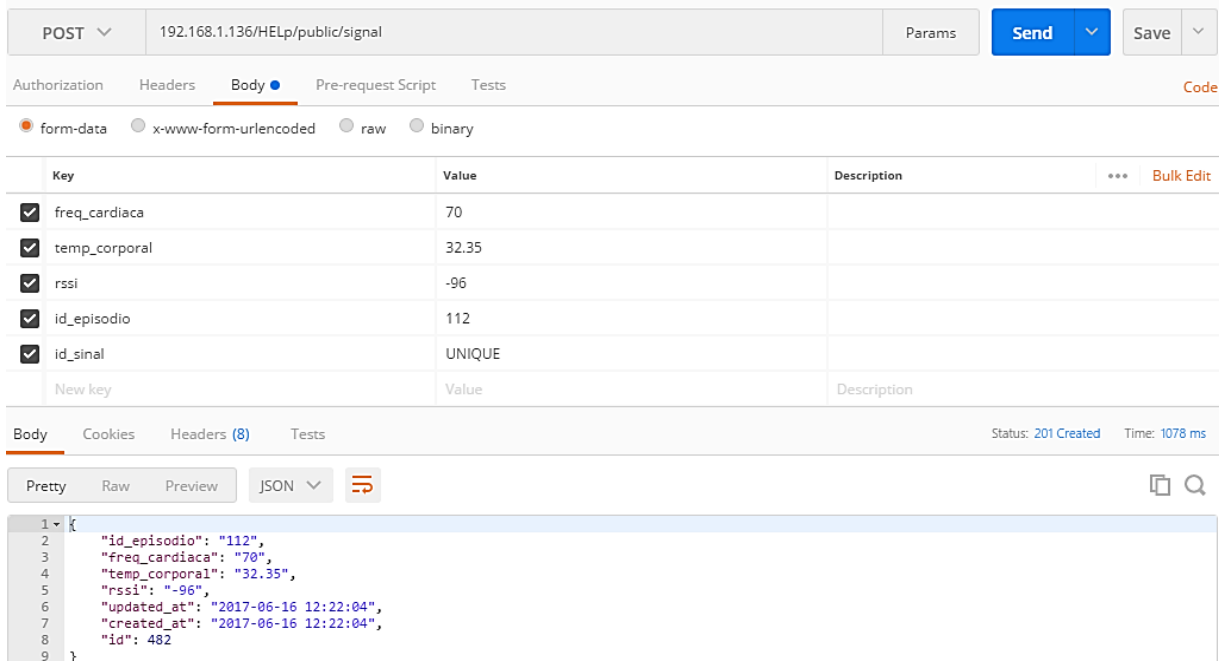


Figura 40 – Interface do *Postman*, com exemplo de um pedido ao servidor e a respetiva resposta.

São, assim, enviados valores para a tabela “*sinais*”, sendo preenchidas todas as suas colunas para um determinado id do sinal, preenchido automaticamente. A resposta é também simples de entender: apenas apresenta o nome da coluna correspondente e o valor que lhe foi atribuído.

Através do *Postman* foi gerado o código *Java* necessário para a aplicação *Android*. Assim, bastou copiar este código e colocá-lo na aplicação *Android* dentro de uma tarefa assíncrona. Através de uma *TimerTask*, esta tarefa pode estar a executar continuamente, ou seja, e neste caso, de cinco em cinco segundos são guardados na base de dados os valores dos parâmetros vitais que estão a ser recolhidos pela pulseira, associados ao ID do episódio em questão. A Figura 41 contém o código *Java* gerado pelo *Postman* para este caso.

```

GENERATE CODE SNIPPETS
Java OK HTTP
Copy to Clipboard

1 OkHttpClient client = new OkHttpClient();
2
3 MediaType mediaType = MediaType.parse("multipart/form-data; boundary
  =----WebKitFormBoundary7MA4YWxkTrZu0gW");
4 RequestBody body = RequestBody.create(mediaType, "
  -----WebKitFormBoundary7MA4YWxkTrZu0gW\r\nContent-Disposition: form-data; name
  =\"freq_cardiaca\"\r\n\r\n70\r\n-----WebKitFormBoundary7MA4YWxkTrZu0gW\r\nContent
  -Disposition: form-data; name=\"temp_corporal\"\r\n\r\n32.35\r\n
  -----WebKitFormBoundary7MA4YWxkTrZu0gW\r\nContent-Disposition: form-data; name
  =\"rssi\"\r\n\r\n-96\r\n-----WebKitFormBoundary7MA4YWxkTrZu0gW\r\nContent-Disposition:
  form-data; name=\"id_episodio\"\r\n\r\n112\r\n
  -----WebKitFormBoundary7MA4YWxkTrZu0gW\r\nContent-Disposition: form-data; name
  =\"id_sinal\"\r\n\r\nUNIQUE\r\n-----WebKitFormBoundary7MA4YWxkTrZu0gW--");
5 Request request = new Request.Builder()
6   .url("http://192.168.1.136/HElp/public/signal")
7   .post(body)
8   .addHeader("content-type", "multipart/form-data; boundary
  =----WebKitFormBoundary7MA4YWxkTrZu0gW")
9   .addHeader("cache-control", "no-cache")
10  .addHeader("postman-token", "1a4dba7b-99f2-c51c-ead-9d88fbede915")
11  .build();
12
13 Response response = client.newCall(request).execute();

```

Figura 41 – Código Java gerado pelo Postman.

Para além do procedimento mencionado, foi apenas necessário importar para o projeto a biblioteca *okhttp* e dar aos valores a enviar o nome das variáveis que armazenam os valores do ID do episódio, a frequência cardíaca e a temperatura corporal.

Deve ter-se em atenção ainda que ambos o servidor e o *smartphone* usados deverão estar ligados à mesma rede, e que o endereço do *localhost* é alterado dependendo da rede ao qual se está ligado, pelo que é necessário alterar o endereço no código da aplicação *Android* antes de a utilizar.

Por fim, é de salientar que na realidade irá ser enviado apenas um valor padrão para a coluna “*rssi*”. O motivo será explicado de seguida, e está relacionado com a localização.

4.6.3.2 Localização

Ao contrário do que acontecia com os parâmetros vitais, o objetivo neste caso não era enviar os dados diretamente da aplicação *Android* para a base de dados, mas sim usar o *web service* implementado pelo LAS para pedir ao servidor qual a localização de determinado utilizador a partir do *JavaScript*. Por sua vez, a resposta do servidor seria captada por uma função *Ajax*, e a posição desenhada no mapa.

Assim, para efetuar o pedido, foi usado o WSDL fornecido para a aplicação, e um *web tester* que permite ver qual a estrutura dos pedidos e as respostas para determinado método. O pedido enviado tem as características referidas para um pedido SOAP, e a resposta é dada em XML. O exemplo da Figura 42 ilustra o pedido e a resposta ao servidor usando o método *getUsersLocation()*. Neste caso, o edifício onde o utilizador se encontra será o Edifício A do IPN, e a resposta retorna os utilizadores guardados na base de dados do servidor, com indicação da sua latitude e longitude.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:GetUsersLocation xmlns:ns2="wifind">
      <arg0>IPN_A</arg0>
    </ns2:GetUsersLocation>
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:GetUsersLocationResponse xmlns:ns2="wifind">
      <return>
        <ListString>IPN_A, IPN_A_F1, Lucia, 40.19246, -8.4146; IPN_A, IPN_A_F1, Teste, 40.1925556233, -8.4147042805;</ListString>
        <UserList>
          <buildingName>IPN_A</buildingName>
          <floorPlanName>IPN_A_F1</floorPlanName>
          <latitude>40.19246</latitude>
          <longitude>-8.4146</longitude>
          <timeStamp>2017-05-29T17:00:00+01:00</timeStamp>
          <userTag>Lucia</userTag>
        </UserList>
        <UserList>
          <buildingName>IPN_A</buildingName>
          <floorPlanName>IPN_A_F1</floorPlanName>
          <latitude>40.1925556233</latitude>
          <longitude>-8.4147042805</longitude>
          <timeStamp>2015-09-05T17:49:20+01:00</timeStamp>
          <userTag>Teste</userTag>
        </UserList>
      </return>
    </ns2:GetUsersLocationResponse>
  </S:Body>
</S:Envelope>
```

Figura 42 – Pedido ao servidor e respetiva resposta usando o método *getUsersLocation()*.

O pedido feito ao servidor é feito de acordo com o apresentado na Figura 42, sendo necessário indicar apenas o tipo de pedido, que mais uma vez é um POST, o nome da aplicação (*WiFind*) e o endereço do servidor em que o *web service* se encontra alojado.

Extraíndo a latitude e longitude que o servidor retorna para cada utilizador, é possível fazer uma conversão, que traduz a posição do utilizador em pixéis, para uma representação no mapa da interface. Esta conversão foi feita a partir de uma interpolação, usando como referência as coordenadas do Edifício A do IPN dadas pelo *Google Maps* e os pixéis de referência do mapa na interface *web*, sendo o algoritmo implementado numa função do *JavaScript*.

Uma vez implementada esta funcionalidade, verificou-se que o sistema implementado pelo LAS apenas tem a capacidade de enviar a posição do utilizador da aplicação *Android*, não guardando os dados da sua posição ao longo do tempo na base de dados. Assim, os

utilizadores retornados na resposta do SOAP são estáticos e representam valores inseridos manualmente na base de dados apenas para efeitos de teste. Uma vez que colocar o sistema a funcionar da forma desejada levaria bastante tempo e exigiria muito trabalho optou-se por, à semelhança do método usado para enviar os valores dos parâmetros vitais recolhidos, usar uma REST API.

O procedimento foi em tudo semelhante ao que tinha sido usado anteriormente, já descrito. Da parte da aplicação *Android* são enviadas a latitude e longitude recebidas pelo servidor e armazenadas numa variável da aplicação, um dos motivos pelo qual não foi usada a coluna “*rssl*” da tabela “*sinais*”. Uma vez que os valores são enviados através de diferentes pedidos em diferentes atividades da aplicação *Android* e que é possível enviar a latitude e longitude de forma individual, a solução mais simples passou por criar uma nova tabela na base de dados que recebe esses dois valores associados ao ID do episódio. Isto evita a necessidade de criar um *array* constituído por estas duas coordenadas na aplicação *Android*, e de posteriormente o “partir” na aplicação *web* para que as duas coordenadas sejam usadas de forma independente, podendo desta forma aceder-se aos valores individualmente. O algoritmo de representação desenvolvido foi usado, uma vez que continua válido para o efeito desejado.

É ainda de referir que também para esta atividade foi criada uma *TimerTask*, com o mesmo intervalo, e que inicia assim que se clica no botão de iniciar a localização *indoor*.

A representação da localização do doente na interface *web* corresponde à Figura 43. De cinco em cinco segundos é executada uma função *JavaScript*, que através do *Ajax* retorna os últimos valores de coordenadas de localização do doente com o ID indicado, guardadas na base de dados.

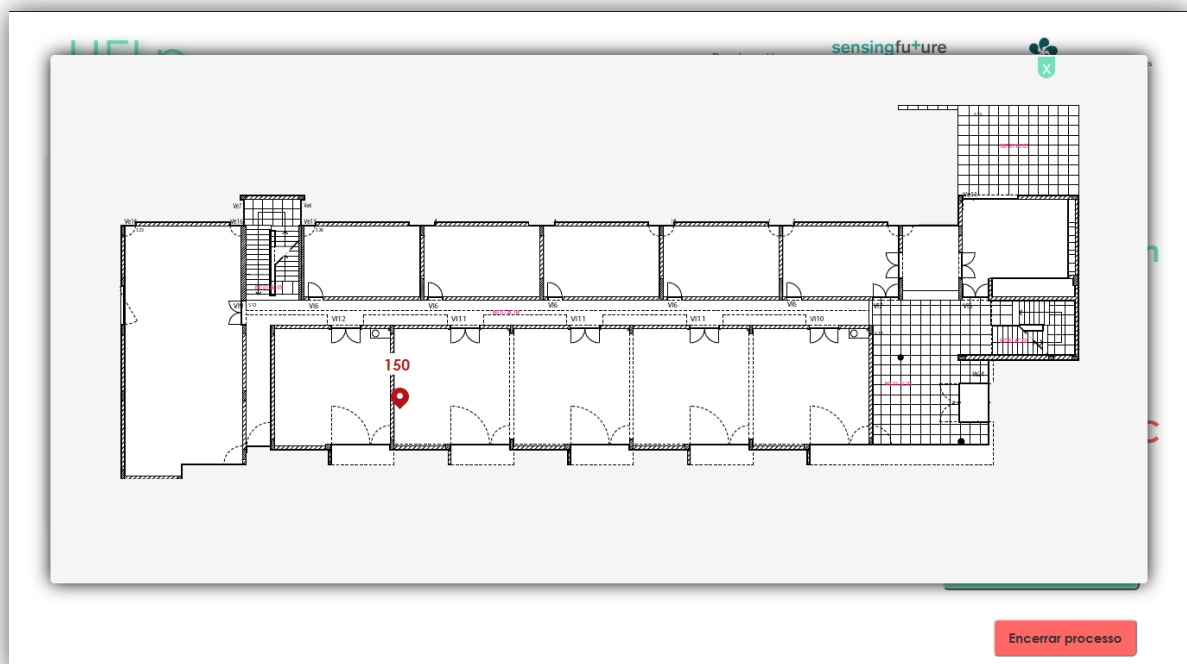


Figura 43 – Última localização do doente 150 no interior do Edifício A do IPN.

Assim, após a implementação das REST API para os parâmetros vitais e para a localização, todas as funcionalidades definidas como obrigatórias e necessárias para que este sistema funcione da forma desejada estão implementadas, e o sistema concluído.

4.7 Considerações finais

Ao longo deste capítulo foram analisadas todas as etapas do desenvolvimento do protótipo, tendo sido enunciada a escolha do *software* e do *hardware* a utilizar, explicado o desenvolvimento de uma aplicação *Android* necessária para comunicar com a pulseira que recolhe os parâmetros vitais do doente, bem como para integrar o algoritmo de localização, e apresentada a implementação da *dashboard* de visualização dos dados. A interação entre estes dois componentes e a base de dados foi também exposta, tendo sido abordados os conceitos de REST API e SOAP.

Relativamente à escolha das *frameworks*, linguagens e bibliotecas a utilizar, foi uma escolha centrada principalmente na portabilidade que estas podem conferir ao sistema desenvolvido, permitindo construir uma aplicação que pode ser acedida, não só através de uma central fixa, mas em qualquer local de um hospital, e em qualquer tipo de dispositivo, sendo apenas necessário o acesso à *Internet*.

Já a decisão de *hardware* teve em conta diversos fatores limitantes, como o custo de aquisição de um dispositivo numa fase inicial, mas mais importante as dificuldades que foram surgindo com as questões de comunicação com o objetivo de recolher os sinais pretendidos. O facto de alguns destes dispositivos terem um protocolo de comunicação fechado ou não documentado torna difícil a interação e a recolha de dados.

Para a implementação da *dashboard* foi essencial um planeamento cuidado da base de dados a implementar, tendo em vista satisfazer os requisitos impostos no Capítulo 3. Identificar as tabelas necessárias e as relações entre elas é essencial para um modelo simples e funcional. Também os modelos conceptual e físico de dados são úteis, na medida em que permitem resumir, não só as relações entre tabelas, como também o tipo de dados de cada coluna, bem como as suas chaves primárias e secundárias, bastante relevantes no desenvolvimento da aplicação. Neste ponto é necessário ter em atenção que os *softwares* utilizados na construção dos modelos criados nunca tinham sido usados pela aluna, tendo sido explorados pela primeira vez. De igual forma, nenhuma das *frameworks*, linguagens e bibliotecas usadas na implementação da *dashboard* tinham antes sido utilizadas pela aluna, o que exigiu bastante tempo de aprendizagem e a realização de diversos tutoriais até se considerar um mínimo de preparação para iniciar a implementação.

Por fim, é de destacar que a utilização de protocolos de comunicação como REST ou SOAP pode representar uma mais-valia para diversas aplicações *web* em que se pretenda transferir dados para serem guardados, ou ainda para serem usados por uma aplicação de outro tipo, uma vez que podem ser usados por diversas linguagens de programação. A facilidade com que podem ser implementados recorrendo a ferramentas como o *Postman* fizeram com que a aluna os conseguisse utilizar de forma eficiente sem ter tido qualquer contacto prévio com este tipo de aplicações.

5. Avaliação do protótipo desenvolvido

Tendo sido já analisados os requisitos inicialmente estabelecidos para este sistema de monitorização e toda a sua implementação, é feita uma descrição global de todos os componentes do protótipo e a forma como comunicam entre si na Secção 5.1. É ainda descrito o seu modo de utilização na Secção 5.2, incluindo-se um breve manual de utilização do sistema no geral, apesar de se dar destaque ao funcionamento da *dashboard*. As conclusões relativas aos testes de usabilidade realizados são expostas na Secção 5.3, que correspondem a uma avaliação do protótipo numa perspetiva de profissionais de saúde. Existiram ainda alguns processos de validação do sistema na perspetiva empresarial, que permitem classificá-lo como *Minimum Viable Product* (MVP), apresentados na Secção 5.4, que aborda também alguns desafios em aberto e possíveis soluções para os ultrapassar. Para finalizar o capítulo, na Secção 5.5 encontram-se as considerações finais relativamente ao trabalho realizado.

5.1 Descrição

O sistema desenvolvido é composto por diversos componentes, cada um útil e necessário para atingir o objetivo principal: a monitorização dos sinais vitais do doente e a sua localização, para que possa ser observado e monitorizado pelo profissional de saúde e seja socorrido atempadamente em caso de agravamento da sua situação clínica, minorando assim eventuais fatalidades. Na Figura 44 estão representados todos os componentes do sistema, que serão descritos e analisados individualmente.

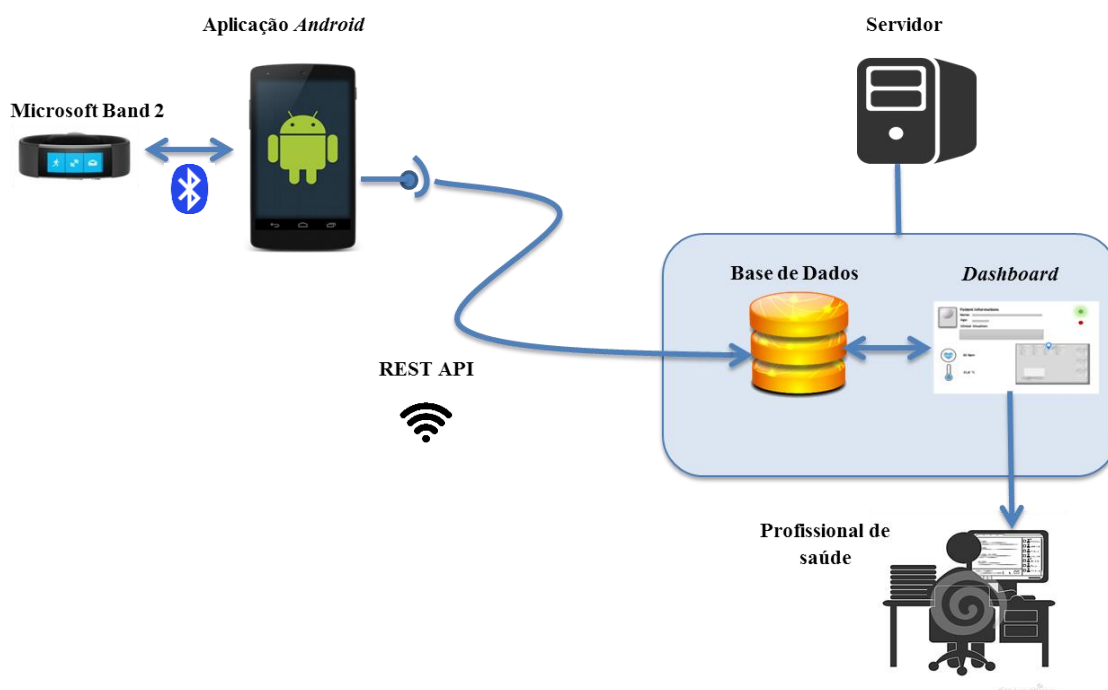


Figura 44 – Componentes do sistema implementado.

Tendo como referência a Figura 44, pode fazer-se a seguinte descrição:

1. A *Microsoft Band 2* é usada para a recolha dos sinais vitais do doente, neste caso a frequência cardíaca e a temperatura corporal. Encontra-se ligada por *Bluetooth* a um *smartphone*.
2. O *smartphone* é necessário para a recolha dos sinais vitais medidos pela pulseira, enviando-os para uma base de dados através de uma REST API, implementada numa aplicação *Android* desenvolvida para a *Microsoft Band 2*. Esta aplicação contém ainda um algoritmo de localização, que permite enviar as coordenadas da localização do utilizador para a base de dados da mesma forma.
3. Os dados enviados pela aplicação *Android* são armazenados numa base de dados alojada num servidor, bem como outros dados inseridos pelo profissional de saúde através da *dashboard*;
4. A *dashboard* comunica com a base de dados, permitindo guardar, atualizar e representar informações relevantes. É ainda capaz de usar os dados da base de dados para gerar alertas.
5. O profissional de saúde tem controlo sobre todo o processo através da *dashboard*, podendo inserir doentes na base de dados, retirar doentes da lista tornando-os inativos e ainda visualizar as informações de cada um, incluindo os alarmes que indicam que existe uma alteração no estado de saúde de um doente.

O modo de utilização do sistema necessário para o seu bom funcionamento é descrito na Secção 5.2.

5.2 Modo de utilização

Para um entendimento correto do modo de utilização descrito é necessário ter em atenção a Figura 45, nomeadamente os botões marcados com um retângulo a vermelho e os campos marcados com um retângulo azul. Todos os elementos essenciais se encontram numerados. As interfaces representadas correspondem às figuras 30, 31, 32 e 33, respetivamente, da esquerda para a direita e de cima para baixo. É ainda necessário ter em atenção o procedimento que traduz o funcionamento da aplicação *Android* desenvolvida, indicado no Capítulo 4.

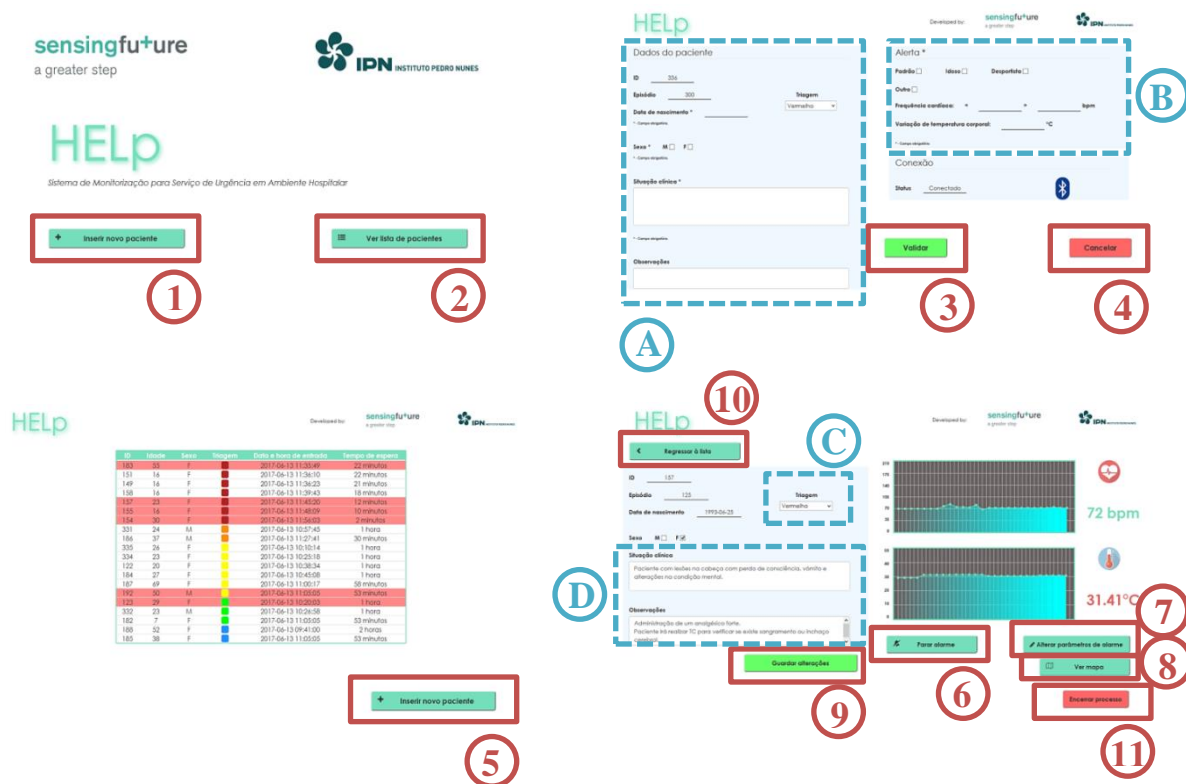


Figura 45 – Interfaces da *dashboard* com botões demarcados a vermelho e campos a azul.

Para utilizar o sistema desenvolvido, é assim necessário proceder da seguinte forma:

1. Na interface inicial, clicar no botão 1 para inserir um novo doente no sistema. Se pretender ver a lista de doentes, clicar no botão 2.
2. Ao inserir um novo doente no sistema, começar por colocar a pulseira no pulso do doente, bem apertada, e verificar que esta se encontra emparelhada com o *smartphone*. Assegurar-se que o *smartphone* está ligado à *Internet*.
3. Preencher a interface com os dados do doente no campo A e definir os parâmetros de alarme no campo B selecionando a opção desejada.
4. Colocar a aplicação *Android* a funcionar de acordo com o procedimento apresentado no Capítulo 4.
5. Verificar que todos os campos obrigatórios foram preenchidos e que não existem erros. Clicar no botão 3 para validar os dados. Para cancelar o registo do novo doente clicar no botão 4.
6. Verificar que o doente foi inserido na lista de doentes.
7. Para inserir um novo doente na lista, clicar no botão 5.
8. Perante a ocorrência de um alarme:
 - a. Clicar no doente com indicação de alarme (linha a vermelho) para ver as suas informações.
 - b. Clicar no botão 6 para parar o alarme.
 - c. Verificar os parâmetros vitais do doente e, se necessário, alterar o seu código de triagem no campo C.
 - d. Alterar os parâmetros de alarme do doente clicando no botão 7.

- e. Localizar o doente clicando no botão 8 para ver o mapa da sala caso este necessite de assistência.
 - f. Se necessário, atualizar a situação clínica do doente e acrescentar as observações necessárias no campo D e guardar as alterações clicando no botão 9.
 - g. Regressar à lista clicando no botão 10.
9. Para encerrar o processo do doente, clicar no botão 11.

5.3 Avaliação do protótipo na perspectiva do profissional de saúde

Com o objetivo de classificar e avaliar o protótipo desenvolvido, nomeadamente no que diz respeito à sua usabilidade, facilidade de utilização, facilidade de aprendizagem e satisfação, foi realizado um teste de usabilidade, descrito na Secção 5.3.1. Os participantes do teste incluem profissionais de saúde de um serviço de urgência e de um serviço de internamentos, bem como engenheiros de *software* para efeitos de comparação de resultados. Os resultados obtidos são discutidos na Secção 5.3.2.

5.3.1 Testes de usabilidade

Usabilidade traduz a facilidade de uso de um objeto de diversos tipos, dos quais fazem parte aplicações de *software* e *websites*. Esta característica é de elevada relevância, uma vez que a facilidade de uso de um *website* é um fator de destaque em relação a possíveis concorrentes, incentivando o utilizador a escolhê-lo em prol de todos os outros por este ser, não apenas agradável e de *design* apelativo, mas também simples de usar e de aprender (Churm, 2012). Assim, os testes de usabilidade pretendem observar um possível utilizador final a interagir com o produto de forma a detetar as dificuldades que este sentiu e possíveis diferenças entre o modo de utilização na visão do programador e o comportamento real do utilizador (“Usability Testing,” n.d.).

Existem vários tipos de questionários para medir a usabilidade de um objeto, alguns deles mais dirigidos à avaliação de *software*. O modelo utilizado para a realização dos testes de usabilidade do protótipo foi o *Usefulness, Satisfaction and Ease of use (USE)* (Dantas et al., 2017). Este modelo prevê um conjunto de trinta perguntas, a serem respondidas de acordo com uma escala de classificação de um a sete pontos, que vão desde “discordo fortemente” a “concordo plenamente”, respetivamente. Este tipo de teste pode ser utilizado em diferentes contextos, desde a avaliação de tecnologias na área da saúde, vídeos, redes sociais, ferramentas de engenharia, realidade virtual, entre outros, sendo portanto bastante genérico (Dantas et al., 2017).

Para a realização do teste de usabilidade tentou-se abranger maioritariamente profissionais de saúde, uma vez que seriam os utilizadores finais do protótipo desenvolvido.

Para efeitos de comparação, o teste foi realizado também por engenheiros. Para as duas classes abrangidas o procedimento do teste foi similar, e consistiu em quatro fases:

1. Foi feita uma breve apresentação do protótipo desenvolvido e ao âmbito em que se pretende que seja utilizado, bem como uma breve explicação acerca do objetivo do teste de usabilidade e como este iria ser realizado;
2. O teste foi realizado com recurso ao manual de utilização apresentado na Secção 5.2 para a *dashboard* e ao modo de utilização da aplicação *Android* apresentado no Capítulo 4;

O participante atribuiu as classificações, de um a sete, às questões do modelo USE, apresentadas na Tabela 10. Na Secção 5.3.2 é apresentada a caracterização dos participantes do teste de usabilidade, bem como uma discussão dos resultados obtidos.

3. Tabela 10;
4. Houve uma pequena discussão sobre a avaliação geral do protótipo, em que o participante pôde colocar questões ou dar algumas sugestões para melhoria do sistema.

Na Secção 5.3.2 é apresentada a caracterização dos participantes do teste de usabilidade, bem como uma discussão dos resultados obtidos.

Tabela 10 – Questões abrangidas pelo modelo USE utilizadas nos testes de usabilidade realizados.

Usabilidade	
1	Ajuda-me a ser mais eficaz.
2	Ajuda-me a ser mais produtivo.
3	É útil.
4	Permite-me um melhor controlo das atividades da minha vida.
5	Torna as coisas que eu quero concretizar mais fáceis de serem feitas.
6	Poupa-me tempo quando o uso.
7	Satisfaz as minhas necessidades.
Facilidade de utilização	
8	Faz tudo o que eu esperaria que fizesse.
9	É fácil de usar.
10	É simples de usar.
11	É intuitivo.
12	Requer o mínimo possível de passos para alcançar o meu objetivo quando o uso.
13	É flexível.
14	O seu uso não requer esforço.
15	Eu consigo usá-lo sem instruções escritas.
16	Eu não noto quaisquer inconsistências quando o uso.
17	Tanto os utilizadores ocasionais como os regulares gostariam dele.
18	Eu consigo corrigir erros de forma rápida e fácil.
19	Eu consigo usá-lo todas as vezes com sucesso.
Facilidade de aprendizagem	
20	Eu aprendi rapidamente a usá-lo.
21	Eu lembro-me facilmente como o usar.
22	É fácil aprender a usá-lo.
23	Eu tornei-me rapidamente competente a usá-lo.
Satisfação	
24	Eu estou satisfeito com o seu uso.
25	Eu recomendá-lo-ia a um amigo.
26	É divertido de usar.
27	Funciona da forma que eu quero que funcione.
28	É maravilhoso.
29	Eu sinto que preciso de tê-lo.
30	É agradável de usar.

5.3.2 Discussão dos resultados

A amostra abrangida é constituída por cinco profissionais de saúde, sendo que dois destes trabalham num serviço de urgência e três num serviço de internamentos. Relativamente à idade dos profissionais de saúde, pode referir-se que o participante mais novo tinha 40 anos e o mais velho 56 anos, o que contrasta com a idade dos engenheiros abrangidos pelo teste, sendo que o mais novo tinha 23 anos e o mais velho 26 anos. A caracterização dos participantes é apresentada na Tabela 11.

Tabela 11 – Caracterização dos participantes do teste de usabilidade realizado.

Profissão (%)	Profissional de saúde	5	(62,5%)
	Engenheiro	3	(37,5%)
Idade dos profissionais de saúde (anos)	Média (SD)	48,2	(1,53)
	Min - Max	40 – 56	
Idade dos engenheiros (anos)	Média (SD)	24,67	(5,85)
	Min - Max	23 – 26	

A média das classificações atribuídas às questões apresentadas no teste de usabilidade encontra-se resumida no gráfico da Figura 46. Através da observação do gráfico verifica-se que não existiu nenhuma questão com avaliação média inferior a 4 pontos, existindo várias questões cuja média é de 7 pontos, pelo que se pode considerar que o sistema foi avaliado de forma bastante positiva.

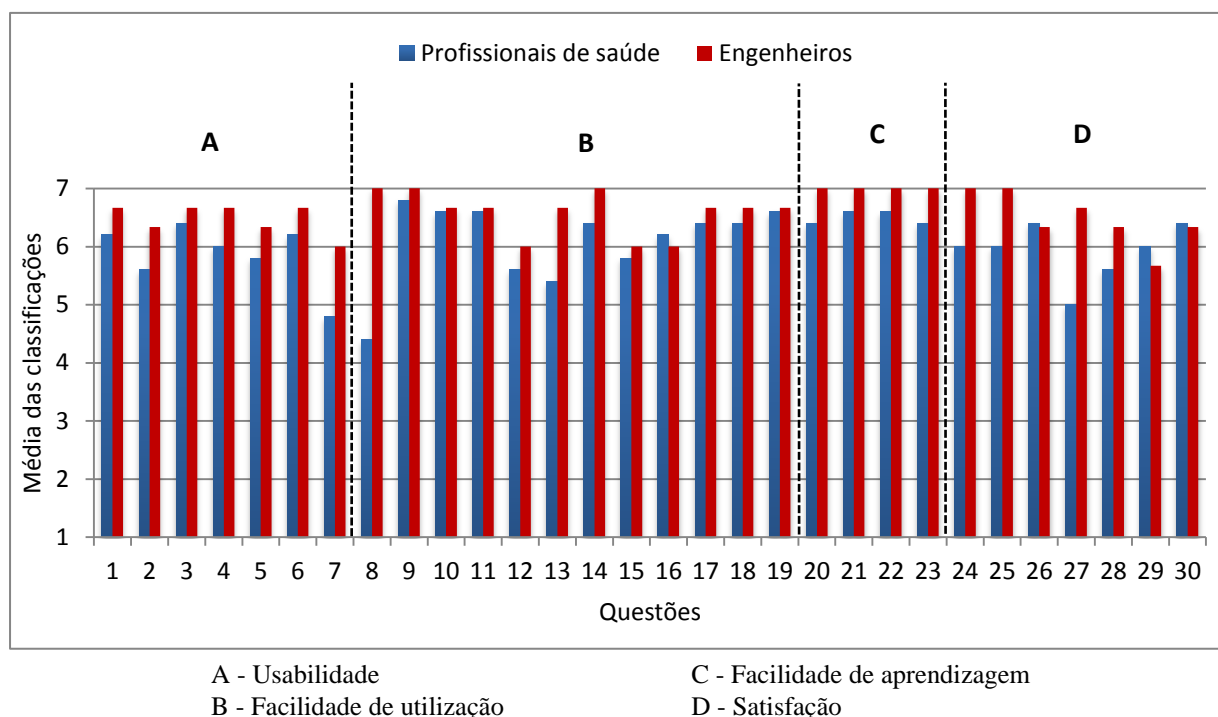


Figura 46 – Média das classificações dadas pelos participantes a cada uma das questões do teste de usabilidade.

No geral, o que se pode observar é que as classificações dadas pelos profissionais de saúde, na maioria das questões, são inferiores às classificações dadas pelos engenheiros. Observa-se ainda que a questão 8 teve a pior classificação dada pelos profissionais de saúde, sendo que as questões 7 e 27 apresentam igualmente classificações médias mais baixas. Na realidade, e após uma pequena discussão com os participantes após a realização do teste, estas classificações podem ser explicadas pelos sinais vitais abrangidos pelo sistema. Na opinião dos profissionais de saúde, os sinais vitais que são medidos pela pulseira poderiam incluir outros sinais vitais, como a saturação do oxigénio no sangue. A diferença que se verifica relativamente à classificação dada às mesmas questões pelo segundo grupo de participantes é que, como engenheiros, não tendo tanto conhecimento da área médica, e por isso uma sensibilidade diferente, mais técnica, avaliam mais o sistema em si e não os parâmetros que são medidos. Recorde-se neste ponto que a decisão acerca dos parâmetros a medir foi uma sugestão da empresa *Glantt*, e que, avaliando o tipo de dispositivos comerciais que existem, incluindo os apresentados na Tabela 6 da Secção 3.6, verifica-se que existem muito poucos dispositivos capazes de medir outros parâmetros vitais para além dos abrangidos por este protótipo, pelo que as soluções de *hardware* existentes constituem um obstáculo no sentido da inclusão de parâmetros vitais a medir pelo sistema.

De forma a avaliar as classificações médias atribuídas a cada uma das quatro categorias do teste de usabilidade (usabilidade, facilidade de utilização, facilidade de aprendizagem e satisfação) construiu-se o gráfico da Figura 47.

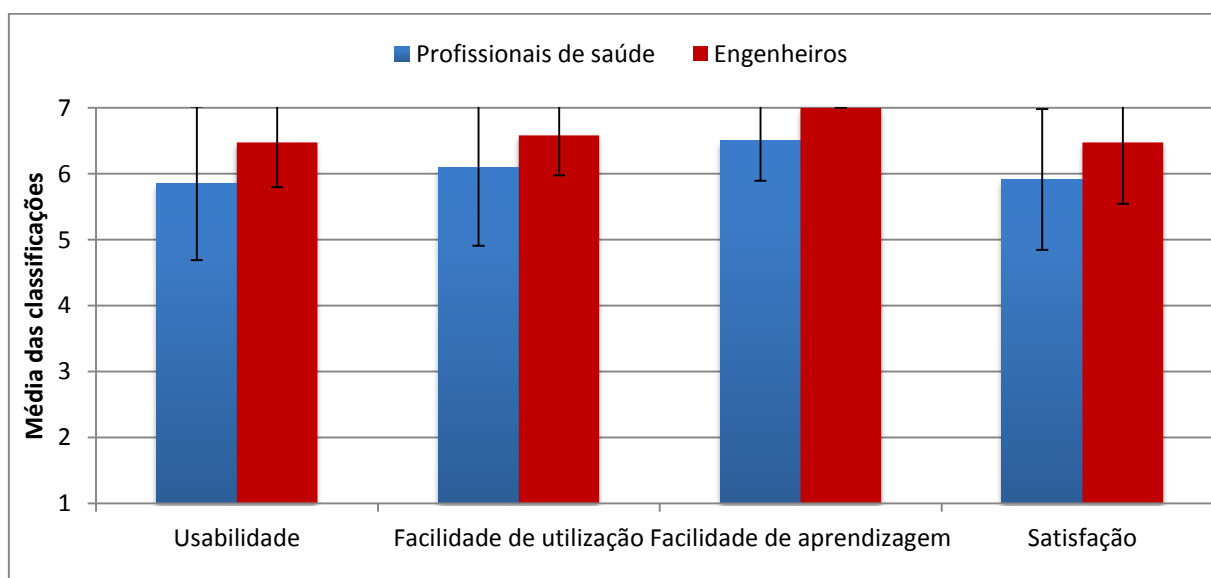


Figura 47 – Média das classificações dadas pelos participantes a cada categoria do teste de usabilidade.

Analisando a categoria de usabilidade, verifica-se que esta é afetada sobretudo pela questão 7, referida anteriormente como uma das questões à qual foi atribuída uma classificação baixa. De forma semelhante, a categoria de facilidade de utilização foi afetada pela classificação atribuída à questão 8; no entanto, a classificação dada pelos dois grupos de participantes é superior a 6, pelo que se pode dizer que o sistema é fácil de utilizar, é intuitivo e não requer esforço na sua utilização. Quanto à categoria que traduz a facilidade de

aprendizagem, no caso do grupo dos engenheiros a média das classificações é de 7 pontos. A diferença relativamente ao grupo dos profissionais de saúde pode ser explicada pela idade dos participantes, que certamente não deverão encarar o uso das novas tecnologias com tanta naturalidade, o que justifica que a sua utilização à primeira vista não tenha sido tão simples e que o sistema tenha de ser usado mais algumas vezes para que se possam tornar competentes a usá-lo. A satisfação pode ser considerada positiva, embora na opinião dos profissionais de saúde pudesse corresponder melhor às expectativas se fosse capaz de medir outros sinais vitais para além dos que são medidos pelo protótipo.

Em suma, o *feedback* recolhido pela aluna relativamente ao protótipo desenvolvido foi bom, ainda que se possam no futuro fazer algumas melhorias, nomeadamente no que diz respeito aos sinais vitais medidos: monitorizando apenas a frequência cardíaca e a temperatura corporal, o sistema não permite ao profissional de saúde acompanhar devidamente o doente. Incluindo a monitorização da saturação do oxigénio no sangue, ou ainda a pressão arterial, que é considerada por estes profissionais um parâmetro relevante, o sistema seria considerado muito bom. A utilização do telemóvel é também um obstáculo sob o ponto de vista dos dois grupos incluídos no teste, uma vez que, para além de aumentar o número de passos necessários ao funcionamento do sistema, não se torna tão intuitivo. Apesar dos pontos em que se poderia melhorar, a *dashboard* é bastante intuitiva e fácil de usar, e seria útil a integração deste sistema no sistema atualmente usado nos serviços de urgência. De acordo com um dos profissionais de saúde inquiridos, seria ainda bastante útil um sistema deste tipo para ser usado em enfermarias: cada enfermaria tem cerca de trinta camas, que durante a noite estão distribuídas por dois ou três enfermeiros, o que dificulta a vigilância constante de todos os doentes, e muitas vezes quem dá o alerta de que um doente se está a sentir mal é o doente que está na cama ao lado. A constante monitorização de parâmetros vitais e a possibilidade de aceder à lista de doentes e aos alertas em diversos ecrãs distribuídos pela enfermaria aumentaria a performance dos profissionais de saúde responsáveis, contribuindo para uma melhoria na qualidade do serviço dos hospitais. O facto de este sistema ser *wireless* também constitui uma vantagem, uma vez que é mais confortável para o doente.

5.4 Avaliação do protótipo na perspetiva empresarial

Com o principal objetivo de avaliar o protótipo desenvolvido foram realizadas duas ações relevantes:

- Foi realizada, em primeiro lugar, uma reunião de apresentação do protótipo, na qual estiveram representados a *Sensing Future Technologies* e o IPN;
- Realizou-se uma apresentação do protótipo à empresa *Glintt*, tendo participado desta reunião um representante da empresa *Glintt*, a *Sensing Future Technologies* e o IPN.

Importa referir que a primeira reunião se centrou mais em aspetos técnicos e nos desafios em aberto, enquanto que a reunião com a empresa *Glintt* dirigiu-se mais para questões de usabilidade e aplicabilidade do ponto de vista do utilizador. A ordem de trabalhos foi semelhante nos dois casos, tendo sido apresentados os objetivos do protótipo

desenvolvido, bem como os componentes que o constituem, dando-se destaque à sua funcionalidade e forma de comunicação, e o seu funcionamento de um modo geral. Após uma demonstração do funcionamento do protótipo teve lugar uma pequena discussão acerca dos desafios em aberto do sistema e de qual a melhor forma de os ultrapassar, tendo sido apresentadas algumas possíveis soluções.

Apesar dos desafios em aberto importa referir que o protótipo desenvolvido foi avaliado como um *Minimum Viable Product*, considerando-se suficiente para um teste em ambiente hospitalar que englobe um número considerável de doentes. Importa também salientar que já existiu uma referência a este sistema por parte da empresa *Glintt* a hospitais, tendo principalmente hospitais privados e grandes grupos públicos dos quais os CHUC manifestado interesse em observar o seu funcionamento. Embora possa não haver confiança total por parte dos médicos nos resultados recolhidos, ter um sistema deste tipo a funcionar no meio hospitalar passa uma imagem de hospital modernizado e informatizado, o que representa uma ideia agradável para algumas entidades. Pode ainda abrir portas para outras aplicações fora dos serviços de urgência.

A necessidade de utilização de um *smartphone* como *gateway* entre a pulseira e o servidor apresentou-se como o maior desafio em aberto do sistema, levantando principalmente questões de preço. A este desafio juntam-se questões como a reutilização e esterilização da pulseira e a sua autonomia, bem como a dificuldade em usar pulseiras de baixo custo devido a protocolos de comunicação fechados e falta de documentação das características *Bluetooth* necessárias para o desenvolvimento. Algumas das sugestões apresentadas como forma de ultrapassar estes desafios foram consideradas inviáveis; no entanto, várias propostas são possíveis de implementar, pelo que devem ser tidas em conta para trabalhos futuros.

Em primeiro lugar, foi sugerida a utilização de um cartão-de-visita com eletrónica capaz de o tornar numa *gateway*, eliminando assim o uso do telemóvel. O sistema de localização pode também ser alterado para usar *beacons* em vez de pontos de acesso *Wi-Fi*, e assim corresponder ao desafio proposto. Usar uma pulseira com comunicação *Wi-Fi* é outra alternativa, ou por outro lado criar um novo dispositivo adequado ao que se pretende, embora ambas as sugestões apresentem limitações, como se pode verificar a partir da Tabela 12. A colaboração com fabricantes de pulseiras de baixo custo surgiu também como alternativa, colmatando-se assim a questão do preço; no entanto, verificou-se inviável, uma vez que não foi mostrado interesse por parte das marcas inquiridas.

Por parte da empresa *Glintt* sugeriu-se aproveitar uma parceria com a *Samsung* para obter algumas pulseiras que permitam demonstrar e testar o protótipo em ambiente hospitalar, podendo existir interesse por parte da marca em patrocinar o projeto. Foi ainda sugerida uma parceria com a *Huawei* com vista ao mesmo objetivo. Colocou-se novamente a questão do desenvolvimento de um novo dispositivo aplicável em braceletes de baixo custo como solução para a reutilização. No entanto, esta solução foi novamente posta de parte, uma vez que seria necessário bastante tempo de desenvolvimento, bem como custos elevados, e poderiam surgir questões de certificação do equipamento. Em alternativa ao uso do telemóvel surgiram três novas soluções: a utilização da tecnologia RFID, sendo necessária a

configuração de uma rede específica; usar recetores *Wi-Fi* ou *Bluetooth*, sendo que usando *Wi-Fi* não seria necessário configurar uma nova rede; usar o telemóvel pessoal do doente que frequenta as urgências, sendo apenas necessário baixar a aplicação *Android* para utilizar o sistema, o que abre portas a outras aplicações. Esta última solução apresenta a desvantagem de não ser acessível a todos devido a questões como a versão do *Android* do dispositivo, ou a utilização do sistema operativo *iOs*.

Por parte da *Sensing Future Technologies*, foi realizado um contacto com a empresa *MediBioSense*, que criou um sistema de monitorização de sinais vitais recorrendo a um adesivo capaz de monitorizar vários parâmetros vitais. Esta solução não inclui um algoritmo de localização, que no entanto poderia vir a ser integrado caso haja interesse. A única limitação seria o preço destes dispositivos, que apesar de serem reutilizáveis, custariam 200€ cada e 40€ por cada adesivo descartável. Por fim, uma outra opção a explorar é o *Bluetooth Secure Gateway Kit*, que recorre a um *Raspberry Pi* para efetuar a comunicação entre dispositivos *Bluetooth* e um servidor, eliminando assim a necessidade de utilização do telemóvel. A Tabela 12 resume as soluções apresentadas e as possíveis limitações que estas podem levantar.

Tabela 12 – Soluções propostas para ultrapassar os desafios em aberto do sistema.

Solução proposta	Objetivo	Possíveis limitações
Uso de um cartão-de-visita que se comporte como uma <i>gateway</i>	Substituir o telemóvel como <i>gateway</i>	Constitui um <i>hardware</i> adicional; passível de ser esquecido em qualquer lugar do hospital
Substituir AP's <i>Wi-Fi</i> por <i>beacons</i>	Substituir o telemóvel como <i>gateway</i>	É necessário inverter o sistema de localização para dispensar o uso do telemóvel
Usar pulseiras com capacidade <i>Wi-Fi</i>	Substituir o telemóvel como <i>gateway</i>	É desconhecida a existência de pulseiras deste tipo capazes de monitorizar sinais vitais
Produção de um novo dispositivo para corresponder às funcionalidades desejadas	Substituir o telemóvel como <i>gateway</i> ; reduzir o preço do sistema	É um processo demorado e de elevados custos; é necessária certificação do dispositivo
Criar colaboração com marca de pulseiras de <i>fitness</i> de baixo custo para haver abertura dos protocolos de comunicação.	Reduzir o preço do sistema	Não foi mostrado interesse por parte da marca inquirida
Parceria com a <i>Samsung</i> ou a <i>Huawei</i>	Patrocínio da marca e cedência de pulseiras para teste; adaptação de pulseiras aos objetivos do projeto	Possível falta de interesse da marca pela área médica
Utilização da tecnologia RFID	Substituir o telemóvel como <i>gateway</i>	É necessária a configuração de uma rede específica
Usar o telemóvel pessoal do doente em vez de adquirir telemóveis	Reduzir o preço do sistema	Podem surgir várias limitações com versões do <i>Android</i> , ou com a utilização de <i>iOs</i> , não sendo uma solução acessível para todos os doentes
Adquirir dispositivos da <i>MediBioSense</i>	Substituir o telemóvel como <i>gateway</i> ; superar desafio de reutilização do dispositivo e esterilização	O equipamento necessita de adaptação para incorporar um algoritmo de localização; o preço do dispositivo é de 200€ e cada adesivo descartável custa 40€
Usar o <i>Bluetooth Secure Gateway Kit</i>	Substituir o telemóvel como <i>gateway</i> ; reduzir o preço do sistema	Esta tecnologia pode não permitir um grande número de dispositivos periféricos ligados em simultâneo

Exploradas as sugestões dadas, pode reduzir-se assim o custo de aquisição deste sistema, o que o torna mais ambicionado por parte dos hospitais, e é ainda aumentado o conforto para o doente quando o utiliza, diminuindo o número de dispositivos acessórios necessários. Apesar de o protótipo ser funcional e facilmente aplicável, pode ainda ser melhorado no futuro a partir das sugestões dadas, sendo possível seguir-se por várias vertentes usando várias tecnologias. A escolha deverá, assim, incidir no menor número de equipamentos possível, na otimização do preço de aquisição e no tempo de desenvolvimento necessário para cada solução.

6. Conclusões e trabalhos futuros

A realidade dos serviços de urgência portugueses, atualmente, enfrenta diversos desafios, de entre os quais se destaca sobretudo um insuficiente número de profissionais de saúde necessários para o pleno funcionamento dos serviços nas épocas de maior afluência por parte dos doentes. O elevado número de doentes que recorre aos serviços de urgência sem apresentar condições de saúde de carácter urgente é por si só um obstáculo, sobrecarregando os profissionais de saúde, ficando estes menos libertos para vigiarem de forma mais atenta os restantes doentes. O sistema desenvolvido tem a intenção de transmitir, tanto ao doente como ao profissional de saúde que o utilizam, uma maior sensação de tranquilidade e segurança, para além de conferir ao hospital onde é utilizado uma imagem de modernidade pela adoção de novas tecnologias para a monitorização contínua e autónoma dos utentes que o frequentam.

Para o desenvolvimento deste protótipo foi fundamental toda a pesquisa desenvolvida em redor dos serviços de urgência, dos protocolos de triagem, dos métodos de monitorização de sinais vitais e de métodos e tecnologias que atualmente podem ser usados na localização *indoor*. Foi ainda levado a cabo um estudo do mercado no que diz respeito a dispositivos de monitorização de parâmetros vitais de forma a selecionar o que melhor correspondia aos requisitos desejados.

A utilização de dispositivos comerciais na monitorização de sinais vitais constitui um desafio, uma vez que não existem muitas soluções com protocolos abertos e com as quais se possa interagir. É ainda de salientar que durante o estágio se tornou necessário explorar vários conceitos desconhecidos, principalmente na área de informática e de metodologias de desenvolvimento de *software*. Para a implementação do protótipo foram exploradas e adquiridas novas competências pela aluna, nomeadamente sobre diversas ferramentas e linguagens de programação, bastante úteis na área da instrumentação biomédica e capazes de conferir características desejáveis a soluções desenvolvidas neste contexto. Também esta aprendizagem foi bastante desafiadora.

O protótipo desenvolvido responde de forma clara aos objetivos propostos. No entanto, não está ainda preparado para ser um sistema comercial devido a alguns desafios em aberto que não puderam ser correspondidos, apesar de o protótipo ter sido avaliado de forma positiva pelos testes de usabilidade realizados e podendo ser considerado um *Minimum Viable Product* na perspetiva empresarial. Estes desafios deverão ser ultrapassados sobretudo para facilitar o uso do sistema, diminuindo o número de passos de configuração necessários e reduzindo o número de dispositivos acessórios ao mínimo possível.

A eliminação do telemóvel como *gateway* entre o dispositivo de monitorização e o servidor constitui o maior desafio a ultrapassar. Atualmente, verifica-se um grande crescimento na área dos *wearables*, tendo surgido diversos dispositivos dirigidos principalmente para aplicações de *fitness*, capazes de monitorizar diversos parâmetros, de entre os quais a atividade e as calorias gastas, em certos casos podendo especificar-se qual o exercício físico praticado, padrões de sono e a frequência cardíaca. Todos estes dispositivos são projetados para uso pessoal, prevendo o uso do telemóvel para ter acesso aos dados recolhidos a partir de uma aplicação *Android* ou *iOs*. O principal objetivo destes dispositivos

não é a monitorização de parâmetros vitais, pelo que apenas alguns são capazes de medir outro sinal vital que não a frequência cardíaca, surgindo assim uma grande dificuldade em encontrar um dispositivo adequado para o protótipo desenvolvido. Da mesma forma, estabelecer um protocolo de comunicação entre o telemóvel e pulseiras de baixo custo, que não são projetadas para serem usadas por programadores, constitui um grande obstáculo ao desenvolvimento. Em alguns casos, o *firmware* não permite o acesso às funcionalidades *Bluetooth* da pulseira; noutros casos, estas funcionalidades não estão documentadas. Assim, como trabalhos futuros, sugerem-se duas vertentes: (1) a tentativa de eliminação do telemóvel como *gateway*, tentando-se usar um único dispositivo capaz de recolher os dados de várias pulseiras em simultâneo; (2) a adaptação de uma pulseira às funcionalidades pretendidas, bem documentada, que funcione ela própria como *gateway*, e assim não necessite de um meio intermediário para a comunicação. Esta última solução pode passar por colaborações com marcas reconhecidas no mercado, com interesse em ter destaque neste tipo de projetos, e que proporcionem dispositivos capazes e de baixo custo para a aplicação em larga escala. Sugere-se ainda a integração da monitorização de outros parâmetros, como a saturação do oxigénio no sangue, que embora seja um desafio pela questão do *hardware* que seria necessário, corresponderia melhor às expectativas dos profissionais de saúde, em particular aqueles que trabalham nos serviços de urgência.

Este estágio permitiu um grande enriquecimento de conhecimentos, tendo existido contacto com novos conceitos, tecnologias, *softwares* e linguagens de programação. De igual forma, e a nível não só profissional mas também pessoal, permitiu desenvolver a capacidade de comunicação e de resolução de problemas, tendo existido o incentivo de conduzir as atividades do estágio com autonomia e espírito crítico. Na realidade, apesar de existirem objetivos específicos definidos no início do estágio, a decisão sobre a forma de os atingir, incluindo o *hardware*, *software* e linguagens a utilizar, ficaram a cargo da aluna e, apesar de algumas funcionalidades do sistema terem sido sugeridas pela empresa *Glintt*, que tem experiência na área do *software* aplicável à saúde, o planeamento de requisitos, funcionalidades e *design* inicial foi feito pela aluna e só posteriormente alterado para ir de encontro às sugestões dadas. Importa ainda referir que o foco deste estágio era atingir um *Minimum Viable Product*, tendo tido essencialmente uma abordagem prática empresarial centrada no desenvolvimento técnico. Por fim, está a ser redigido um artigo científico sobre o trabalho desenvolvido, sendo expectável a sua submissão a curto prazo a um congresso científico internacional da área.

Bibliografia

- Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., Vrana, J. (2017). PHP Manual. PHP Documentation Group. Retrieved from <https://secure.php.net/manual/en/preface.php>
- Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. A., & Al-Khalifa, H. S. (2016). Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors*, *16*(5), 707. <https://doi.org/10.3390/s16050707>
- Allegaert, K., Casteels, K., Gorp, I. van, & Bogaert, G. (2014). Tympanic, Infrared Skin, and Temporal Artery Scan Thermometers Compared with Rectal Measurement in Children: A Real-Life Assessment. *Current Therapeutic Research*, *76*, 34–38. Retrieved from [http://www.currenttherapeuticres.com/article/S0011-393X\(13\)00035-0/fulltext](http://www.currenttherapeuticres.com/article/S0011-393X(13)00035-0/fulltext)
- Alliance, Z. (2009). ZigBee Wireless Sensor Applications for Health , Wellness and Fitness, (March), 1–15. Retrieved from <https://docs.zigbee.org/zigbee-docs/dcn/09?4962.pdf> (last accessed April 20, 2013)
- Bautista, N. (2012). Build Web Apps from Scratch with Laravel - The Eloquent ORM. Retrieved April 24, 2017, from <https://code.tutsplus.com/tutorials/build-web-apps-from-scratch-with-laravel-the-eloquent-orm--net-25631>
- Bell, D. (2004). The sequence diagram. IBM. Retrieved from <http://www.ibm.com/developerworks/rational/library/3101.html>
- Bernhardt. (2008, March). Temperature Measurement for Patients with Fever. *33*(3), 14–17. Retrieved from <https://www.uspharmacist.com/article/temperature-measurement-for-patients-with-fever>
- Biler, D. (2013). How to install Laravel on WAMP for beginners. Retrieved April 12, 2017, from <http://www.darwinbiler.com/how-to-install-laravel-on-wamp-for-beginners/>
- Blaz. (2015). Indoor Positioning Systems based on BLE Beacons – Basics. Retrieved March 29, 2017, from <https://locatify.com/blog/indoor-positioning-systems-ble-beacons/>
- Bluetooth Low Energy Technology. (n.d.). Retrieved March 29, 2017, from <http://www.connectblue.com/about-us/technologies/bluetooth-low-energy-technology/>
- Bonato, P. (2005). Advances in wearable technology and applications in physical medicine and rehabilitation. *Journal of Neuroengineering and Rehabilitation*, *2*, 2. <https://doi.org/10.1186/1743-0003-2-2>
- Bonsor, K., & Fenlon, W. (2007). How RFID Works. Retrieved March 29, 2017, from <http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid1.htm>
- Bootstrap Grid System. (n.d.). Retrieved March 23, 2017, from https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp
- Borges, L. (2016, February 1). Já é possível consultar o tempo de espera nas urgências. *Jornal de Negócios*. Retrieved from http://www.jornaldenegocios.pt/economia/saude/detalhe/ja_e_possivel_consultar_tempo_de_espera_para_consulta_nas_urgencias
- Boswell, W. (2016). What is HTML? Retrieved March 17, 2017, from <https://www.lifewire.com/what-is-html-3482374>
- Braun, S. R. (1990). Respiratory Rate and Pattern. In Butterworths (Ed.), *Clinical Methods: The History, Physical, and Laboratory Examinations* (3rd ed.). Boston. Retrieved from <https://www.ncbi.nlm.nih.gov/books/NBK365/>
- Bringing MySQL to the web. (n.d.). Retrieved April 12, 2017, from <https://www.phpmyadmin.net/>
- Britannica, T. E. of E. (2016). Wi-Fi. Retrieved January 24, 2017, from <https://www.britannica.com/technology/Wi-Fi>
- Brown, P. (2013a). Routing in Laravel 4. Retrieved April 26, 2017, from <http://culttt.com/2013/08/26/routing-laravel-4/>
- Brown, P. (2013b). Setting up your first Laravel 4 Controller. Retrieved April 24, 2017, from

- <http://culttt.com/2013/07/01/setting-up-your-first-laravel-4-controller/>
- Brown, P. (2013c). What are RESTful Web Services? Retrieved April 24, 2017, from <http://culttt.com/2013/04/15/what-are-restful-web-services/>
- Caelum. (n.d.). Introdução a PHP. Retrieved March 17, 2017, from <https://www.caelum.com.br/apostila-html-css-javascript/introducao-a-php/>
- Carl Bianco. (2000). How Emergency Rooms Work. Retrieved December 21, 2016, from <http://people.howstuffworks.com/emergency-room2.htm>
- Cariço, M. (2017). Enchente nas urgências hospitalares. Porque é que isto se repete? E como se resolve? *Observador*, (483). Retrieved from <http://observador.pt/especiais/enchente-nas-urgencias-hospitalares-porque-e-que-isto-se-repete-e-como-se-resolve/>
- Chalari, E., Intas, G., Stergiannis, P., Vezyridis, P., Paraskevas, V., & Fildissis, G. (2012). The importance of vital signs in the triage of injured patients. *Critical Care Nursing Quarterly*, 35(3), 292–8. <https://doi.org/10.1097/CNQ.0b013e318255d6b3>
- Chang, P. (2017). [MySQL] Note: Relation: 1–1, 1-n, n-n, nest. Retrieved May 10, 2017, from <https://hackernoon.com/mysql-tutorial-example-relation-foreign-key-database-funtion-join-table-query-one-namy-nest-41dd09648fbd>
- Cheriyedath, S. (2016). Photoplethysmography (PPG). Retrieved December 15, 2016, from [http://www.news-medical.net/health/Photoplethysmography-\(PPG\).aspx](http://www.news-medical.net/health/Photoplethysmography-(PPG).aspx)
- Childs, C., Harrison, R., & Hodkinson, C. (1999). Tympanic membrane temperature as a measure of core temperature. *Arch Dis Child*, 80, 262–266. Retrieved from <http://adc.bmj.com/content/80/3/262.full.pdf+html>
- Chirchir, E. (n.d.). How To Use A Laravel Model – Eloquent ORM. Retrieved April 24, 2017, from <http://simpledeveloper.com/how-to-use-laravel-model/>
- Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). Web Services Description Language (WSDL) 1.1. Retrieved June 14, 2017, from https://www.w3.org/TR/wsdl#_document-s
- Christensson, P. (2013). WAMP Definition. Retrieved April 12, 2017, from <https://techterms.com/definition/wamp>
- Circuit Cellar Staff. (2014). Triangulation, Trilateration, or Multilateration? Retrieved March 30, 2017, from <http://circuitcellar.com/ee-tips/triangulation-trilateration-or-multilateration-ee-tip-125/>
- Clemente, M. P. (2012). Futuro do Empreendedorismo Médico com relevância nas radiofrequências aplicadas à Biotecnologia. In *6º congresso do comité português da URSI*. Lisboa. Retrieved from https://www.anacom.pt/streaming/PaisClemente_CongressoURSI2012.pdf?contentId=1147248&field=ATTACHED_FILE
- Cm. (2015). Centro Hospitalar abre inquérito a morte nas urgências. Retrieved from http://www.cmjornal.pt/sociedade/detalhe/centro_hospitalar_abre_inquerito_a_morte_nas_urgencias
- Colantonio, J. (2017). 14 Open Source API Testing Tools For REST & SOAP Services. Retrieved June 14, 2017, from <https://www.joecolantonio.com/2017/05/16/12-open-source-api-testing-tools-rest-soap-services/>
- Composer - Introduction. (n.d.). Retrieved April 11, 2017, from <https://getcomposer.org/doc/00-intro.md>
- Creating a Basic ToDo Application in Laravel 5 – Part 1. (2015). Retrieved April 24, 2017, from <https://www.flynsarmy.com/2015/02/creating-a-basic-todo-application-in-laravel-5-part-1/>
- CSS Newbies. (2015). Using CSS with PHP Tutorial. Retrieved March 17, 2017, from <http://www.cssnewbie.com/using-css-with-php-tutorial/>
- Danckers, M. (2016). Arterial Blood Gas Sampling. Retrieved January 3, 2017, from <http://emedicine.medscape.com/article/1902703-overview#a1>

- Dantas, C., Jegundo, A. L., Quintas, J., Martins, A. I., Queirós, A., & Rocha, N. P. (2017). European Portuguese Validation of Usefulness, Satisfaction and Ease of Use Questionnaire (USE). In Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, & S. Costanzo (Eds.), *Recent Advances in Information Systems and Technologies: Volume 2* (pp. 561–570). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-56538-5_57
- Daw, W., Kingshott, R., Saatchi, R., Burke, D., Holloway, A., Travis, J., ... Elphick, H. (2016). Medical Devices for Measuring Respiratory Rate in Children : a Review. *Journal of Advances in Biomedical Engineering and Technology*, 3(0), 21–27.
- DeMoes, A. F. A. (2016). *AirLoc : Pedestrian dead reckoning for passenger localization*. Delft University of Technology. Retrieved from <http://www.es.ewi.tudelft.nl/msc-theses/2017-deMoes.pdf>
- Dhingra, S. (2016). REST vs. SOAP: Choosing the best web service. Retrieved June 14, 2017, from <http://searchmicroservices.techtarget.com/tip/REST-vs-SOAP-Choosing-the-best-web-service>
- Dias, O. (2014). Laravel: Rotas e Controllers. Retrieved April 24, 2017, from <https://magazine.softerize.com.br/tutoriais/php/laravel/laravel-rotas-e-controllers>
- Difference Between Heart Rate and Pulse. (n.d.). Retrieved December 15, 2016, from http://support.polar.com/en/support/Difference_Between_Heart_Rate_and_Pulse
- ECG Wireless Hospital Monitoring Solution. (2016). Retrieved January 12, 2017, from <http://intelebens.com/hospital-monitoring>
- Egan, D. (2016). Generate URLs to Named Routes in Laravel. Retrieved April 26, 2017, from <http://dev-notes.eu/2016/11/generate-urls-to-named-routes-in-laravel/>
- Elliott, M., & Coventry, A. (2012). Critical care: the eight vital signs of patient monitoring. *British Journal of Nursing*, 21(10), 621–625. <https://doi.org/10.12968/bjon.2012.21.10.621>
- Estimote. (n.d.). Estimote Indoor Location. Retrieved March 31, 2017, from <https://estimote.com/indoor/>
- Expresso. (2015). Falhas nas urgências. “Só não foram noticiadas mais mortes porque as famílias, provavelmente, não se aperceberam.” Expresso. Retrieved from <http://expresso.sapo.pt/sociedade/falhas-nas-urgencias-so-nao-foram-noticiadas-mais-mortes-porque-as-familias-provavelmente-nao-se-aperceberam=f904914>
- Foley, M. (2007, November). How does Bluetooth work? *Scientific American*. Retrieved from <https://www.scientificamerican.com/article/experts-how-does-bluetooth-work/>
- Gao, T., Greenspan, D., Welsh, M., Juang, R. R., & Alm, A. (2005). Vital Signs Monitoring and Patient Tracking Over a Wireless Network. *27th Annual International Conference of the IEEE EMBS*, 27(September). Retrieved from <https://pdfs.semanticscholar.org/4716/d634a6d138b87d59981106fde9b3d103df61.pdf>
- Garcia-marques, T. (2006). Qual a fiabilidade do alfa de Cronbach? *Questões antigas e soluções modernas ?*, 4(1), 65–90.
- Garcia, G. O., Nunez, S. I. M., Saavedra, L. A., & Tapia, M. D. I. (2010). Real time location system with ZigBee technology. Retrieved from <https://www.google.pt/patents/EP2196816A1?cl=en&dq=zigbee+indoor+location+system&hl=pt-PT&sa=X&ved=0ahUKEwial1ty4hoHTAhUB7hoKHTYsBfi4ChDoAQghMAE>
- Gast, M. (2005). *802.11 wireless networks : the definitive guide*. O’Reilly.
- Gaudlitz, E. (2015). Indoor navigation using WiFi as a positioning technology. Retrieved March 30, 2017, from <https://www.infsoft.com/blog-en/articleid/40/indoor-navigation-using-wifi-as-a-positioning-technology>
- Getting started with the REST API. (n.d.). Retrieved June 14, 2017, from <https://developer.linkedin.com/docs/rest-api#>

- Gilboy, N., Tanabe, P., Travers, D. A., Rosenau, A. M., & Eitel, D. R. (2005). Index , Version 4 : Implementation. *Agency for Healthcare Research and Quality*, 5-46–2.
- Gomes, M. M. de O. R. (2013). *O perfil do utilizador abusivo dos Serviços de Urgência Portugueses: um estudo no Hospital de São João*. Retrieved from http://recipp.ipp.pt/bitstream/10400.22/4546/1/DM_MafaldaGomes_2014.pdf
- Griffiths, P., Saucedo, A. R., Schmidt, P., & Smith, G. (2015). Vital signs monitoring in hospitals at night. *Nursing Times*, 111(36–37), 16–7. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/26434188>
- Harnwood, M. (2009). *Network+ Exam Cram: Wireless Networking*. Retrieved January 18, 2017, from <http://www.pearsonitcertification.com/articles/article.aspx?p=1329709&seqNum=4>
- Haymond, S. (2006). Oxygen Saturation - A Guide to Laboratory Assessment. *Clinical Laboratory News*.
- He, S., & Chan, S. H. G. (2016). Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys and Tutorials*, 18(1), 466–490. <https://doi.org/10.1109/COMST.2015.2464084>
- Hernandez, O., & Jain, V. (2009). Position Location Monitoring Using IEEE® 802.15.4/ZigBee® technology. *Beyond Bits, Issue ...*, 67–73. Retrieved from <http://www.freescale.com/files/microcontrollers/doc/brochure/PositionLocationMonitoring.pdf>
- Hospitais a rebentar pelas costuras. (2014, October). *DECO Proteste*, 33–35.
- How does Bluetooth work? (2007). Retrieved January 27, 2017, from <https://www.scientificamerican.com/article/experts-how-does-bluetooth-work/>
- Humbrecht, R. (2016). What's a Website Mockup? Retrieved June 22, 2017, from <https://www.thoughtspacedesigns.com/blog/post/whats-a-website-mockup/>
- Hutson, A. (2016). Cassandra Conceptual Data Modeling. Retrieved March 15, 2017, from <http://datascale.io/cassandra-conceptual-data-modeling/>
- Ijaz, F., & Yang, H. (2013). Indoor positioning: A review of indoor ultrasonic positioning systems. *Advanced Communication Technology (ICACT)*, 1146–1150. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6488379
- Índice de Saúde Sustentável. (2016) (pp. 1–36). Retrieved from http://www.tsf.pt/Galerias/PDF/2016/03/estudo_abbvie.pdf
- Indoor location technologies compared. (2015). Retrieved January 13, 2017, from <http://lighthouse.io/indoor-location-technologies-compared/>
- Indoor Positioning, Tracking and Indoor Navigation with Beacons. (n.d.). Retrieved March 29, 2017, from <https://www.infsoft.com/technology/sensors/bluetooth-low-energy-beacons>
- IndoorAtlas. (n.d.). IndoorAtlas - Making Indoor Worlds Discoverable. Retrieved March 31, 2017, from <http://www.indooratlas.com/>
- Instituto Pedro Nunes. (n.d.). Laboratório de Automática e Sistemas. Retrieved June 21, 2017, from <https://www.ipn.pt/laboratorio/LAS>
- John Hopkins Medicine. (n.d.). Vital Signs (Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure). Retrieved December 7, 2016, from http://www.hopkinsmedicine.org/healthlibrary/conditions/cardiovascular_diseases/vital_signs_body_temperature_pulse_rate_respiration_rate_blood_pressure_85,P00866/
- K.Kaemarungsi, & P.Krishnamurthy. (2004). Modeling of Indoor Positioning Systems Based on Location Fingerprinting. *Joint Conference of the IEEE Computer and Communications Societies*, 2(C), 1012–1022. <https://doi.org/10.1109/INFCOM.2004.1356988>
- Kárník, J., & Streit, J. (2016). Summary of available indoor location techniques. *IFAC-PapersOnLine*, 49(25), 311–317. <https://doi.org/10.1016/j.ifacol.2016.12.055>

- Kazankov, V. (2016). The Top 10 Advantages Of Using Laravel PHP Framework. Retrieved March 17, 2017, from <https://belitsoft.com/laravel-development-services/10-benefits-using-laravel-php-framework>
- Kelly, B. B. (2014). Emergency Medicine : What to Expect as a Patient in the “ ER .” Retrieved from http://www.sturdymemorial.org/images/pressreleases/ECC_The_ER_Experience_Column_10.13.pdf
- Khan, Y., Ostfeld, A. E., Lochner, C. M., Pierre, A., & Arias, A. C. (2016). Monitoring of Vital Signs with Flexible and Wearable Medical Devices. *Advanced Materials*, 28(22), 4373–4395. <https://doi.org/10.1002/adma.201504366>
- Kornuta, C., Acosta, N., & Toloza, J. (2013). Indoor Positioning Using the Modified Fingerprint Technique. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(10), 2282–2290. Retrieved from https://www.ijirccce.com/upload/2013/december/2_Indoor.pdf
- Koyuncu, H., & Yang, S. H. (2010). A Survey of Indoor Positioning and Object Locating Systems. *International Journal of Computer Science and Network Security (IJCSNS '10)*, 10(5), 121–128. Retrieved from <https://pdfs.semanticscholar.org/9304/f80fbca5bfdcc3820543d186d2b2da5b1c4d.pdf>
- Kumar, S. (n.d.). Efficient Heart Rate Monitoring. *Cypress Semiconductor Corp.*
- Kumar, S. (2015). Difference Between Library and Framework. Retrieved March 17, 2017, from <http://www.c-sharpcorner.com/UploadFile/a85b23/framework-vs-library/>
- Lan, K., & Shih, W. (2014). Using Smart-Phones and Floor Plans for Indoor Location Tracking, 44(2), 211–221.
- Laravel. (n.d.-a). Laravel - Introduction. Retrieved March 17, 2017, from <https://laravel.com/docs/4.2/introduction>
- Laravel. (n.d.-b). Laravel - Migrations & Seeding. Retrieved April 24, 2017, from <https://laravel.com/docs/4.2/migrations>
- Level, S., Mitra, I. T., It, E., & Ibm, A. (2008). Documenting software architecture, Part 1: What software architecture is, and why it’s important to document it. *Discover*, Bachmann, F., Bass, L., Clements, P., Garlan, D.,. Retrieved from <https://www.ibm.com/developerworks/library/ar-archdoc1/ar-archdoc1-pdf.pdf>
- Ljunggren, M., Castrén, M., Nordberg, M., & Kurland, L. (2016). The association between vital signs and mortality in a retrospective cohort study of an unselected emergency department population. *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine*, 24(21), 1–11. <https://doi.org/10.1186/s13049-016-0213-8>
- Lopes, V. (2011). Estruturação de uma Base de Dados. pplware. Retrieved from <https://pplware.sapo.pt/tutoriais/estruturacao-de-uma-base-de-dados/>
- Luo, Y., Hoerber, O., & Chen, Y. (2013). Enhancing Wi-Fi fingerprinting for indoor positioning using human-centric collaborative feedback. *Human-Centric Computing and Information Sciences*, 3(1), 2. <https://doi.org/10.1186/2192-1962-3-2>
- Lyon, R. M., Walker, R. G., Edwards, J. S., Andersson, N., & Johnson, B. J. (2013). Patient monitoring device with remote alert. United States.
- Marquez, L., & Ballen, M. (1945). *Best Practice*, 3(3), 1–6.
- McCombie, D., Dhillon, M., Banet, M., Trommer, G., & Moon, J. (2009). Graphical “mapping system” for continuously monitoring a patient’s vital signs, motion, and location. United States: IMEKO.
- Microsoft. (n.d.-a). Introdução às tabelas. Microsoft. Retrieved from <https://support.office.com/pt-pt/article/Introdução-às-tabelas-03f58e81-86cd-46ad-8199-4122152c7eff>
- Microsoft. (n.d.-b). Noções básicas da base de dados. Retrieved March 14, 2017, from <https://support.office.com/pt-pt/article/Noções-básicas-da-base-de-dados-a849ac16->

07c7-4a31-9948-3c8c94a7c204

- Mike Cohn. (2004). Advantages of User Stories for Requirements. Retrieved February 22, 2017, from <https://www.mountaingoatsoftware.com/articles/advantages-of-user-stories-for-requirements>
- Mitchell, B. (2016). UWB - Ultra-Wide Band. Retrieved March 29, 2017, from <https://www.lifewire.com/ultra-wide-band-817953>
- Mountain Goat Software. (n.d.). User Stories. Retrieved February 22, 2017, from <https://www.mountaingoatsoftware.com/agile/user-stories>
- Mozilla Developer Network and individual contributors. (2017). Ajax - Getting Started. Retrieved March 17, 2017, from https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started
- Mueller, J. (2013). Understanding SOAP and REST Basics And Differences. Retrieved June 14, 2017, from <https://blog.smartbear.com/apis/understanding-soap-and-rest-basics/>
- MySNS Tempos - A Nova Versão da App Já Disponível. (n.d.). Retrieved November 23, 2016, from <http://spms.min-saude.pt/product/aplicacoes-moveis-do-sns/>
- Nield, D. (2016). What is Bluetooth? Retrieved January 27, 2017, from <http://www.techradar.com/how-to/computing/what-is-bluetooth-1323284>
- Noabeb, L. (2010). Anatomy of a Web Service: XML, SOAP and WSDL for Platform-independent Data Exchange. Retrieved from http://www.webreference.com/authoring/web_service/index.html
- NXP Laboratories UK. (2012). ZigBee PRO Stack User Guide (Rev. 2.5), (December), 418.
- Optimus Information. (2016). Traditional vs Agile Software Development. Retrieved February 22, 2017, from <http://www.optimusinfo.com/traditional-vs-agile-software-development/>
- Papapostolou, A., & Chaouchi, H. (2011). RFID-assisted indoor localization and the impact of interference on its performance. *Journal of Network and Computer Applications*, 34(3), 902–913. <https://doi.org/10.1016/j.jnca.2010.04.009>
- Ponniah, P. (2007). *Data Modeling Fundamentals: A Practical Guide for IT Professionals*. Hoboken: John Wiley & Sons, Inc. Retrieved from https://books.google.pt/books?id=VszO_kZl4bUC&printsec=frontcover&hl=pt-PT
- Poole, I. (n.d.-a). IEEE 802.11a. Retrieved January 18, 2017, from <http://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11a.php>
- Poole, I. (n.d.-b). RFID Frequencies and Frequency Bands. Retrieved March 29, 2017, from <http://www.radio-electronics.com/info/wireless/radio-frequency-identification-rfid/readers-writers-printers.php>
- Poolet, M. A. (1999). SQL by Design: Why You Need Database Normalization. SQL Server Pro. Retrieved from <http://sqlmag.com/database-performance-tuning/sql-design-why-you-need-database-normalization>
- Portugal - Manchester Triage System. (n.d.). Retrieved November 23, 2016, from <http://www.triagenet.net/en/node/1>
- Proximi.io. (n.d.). Proximi.io - Build apps that react to the physical world. Retrieved March 31, 2017, from <https://proximi.io/>
- Quaine, N. (n.d.). SOAP Basics. Retrieved June 14, 2017, from <http://www.soapuser.com/basics1.html>
- Quintas, J., Cunha, A., Serra, P., Pereira, A., & Marques, B. (2013). Indoor Localization and Tracking Using 802 . 11 Networks and Smartphones, 117–118.
- Rasmusson, J. (n.d.). What is Agile? Retrieved February 22, 2017, from http://www.agilenutshell.com/how_does_it_work
- Rees, D. (2014). Code Bright: Basic Routing. Retrieved April 26, 2017, from <https://daylerees.com/codebright-basic-routing/>
- RESTful Web Services - Introduction. (n.d.). Retrieved June 13, 2017, from

- https://www.tutorialspoint.com/restful/restful_introduction.htm
- Reyes, J. (2009). Discussing PHP Frameworks: What, When, Why and Which? Retrieved March 17, 2017, from <https://www.noupe.com/development/discussing-php-frameworks.html>
- Reyes, O. C. (2014). Dynamic WIFI Fingerprinting Indoor Positioning System, (October). RFID Technology Primer. (n.d.). Retrieved March 29, 2017, from <http://www.impinj.com/resources/about-rfid/how-do-rfid-systems-work/>
- Rodriguez, A. (2015). RESTful Web services: The basics. Retrieved June 13, 2017, from http://www.ibm.com/developerworks/data/library/dmmag/DMMag_2009_Issue2/Industry/index.html
- Rosenblum, D. S., & Science, C. (2001). UML Component Diagrams and Software Architecture— Experiences from the W. *Interface*, 1–4. Retrieved from <https://pdfs.semanticscholar.org/bbc6/6955f6c88cfa7aa1f4d0c89b30e81acb61f9.pdf>
- Rouse, M. (2005). entity-relationship model (ERM or ER model). TechTarget. Retrieved from <http://searchsqlserver.techtarget.com/definition/entity-relationship-model>
- Rouse, M. (2008). Ultra wideband. Retrieved March 29, 2017, from <http://whatis.techtarget.com/definition/ultra-wideband>
- Rouse, M. (2016). RESTful API. Retrieved June 13, 2017, from <http://searchcloudstorage.techtarget.com/definition/RESTful-API>
- Rouse, M. (2017). database (DB). TechTarget. Retrieved from <http://searchsqlserver.techtarget.com/definition/database>
- Schmidt, P. E., Meredith, P., Prytherch, D. R., Watson, D., Watson, V., Killen, R. M., ... Smith, G. B. (2015). Impact of introducing an electronic physiological surveillance system on hospital mortality. *BMJ Quality & Safety*, 24(1), 10–20. <https://doi.org/10.1136/bmjqs-2014-003073>
- Senion. (2016). *Indoor Positioning 101 - Whitepaper*. Retrieved from <https://senion.com/>
- Sense and Transmit. (2017). Retrieved January 12, 2017, from <http://www.equival.co.uk/products/tnr/sense-and-transmit>
- Sensing Future Technologies. (n.d.). Sensing Future Technologies. Retrieved June 21, 2017, from <http://www.sensingfuture.pt/index.php/pt/empresa>
- Sharma, S. (2015). 15 Best PHP Libraries Every Developer Should Know. Retrieved March 17, 2017, from <https://www.programmableweb.com/news/15-best-php-libraries-every-developer-should-know/analysis/2015/11/18>
- Smith, J. (1992). Understanding pulse oximetry. *Anaesthesia and Intensive Care*.
- Suehring, S., & Valade, J. (n.d.). How to Organize the Data for Your MySQL Database. dummies. Retrieved from <http://www.dummies.com/programming/databases/how-to-organize-the-data-for-your-mysql-database/>
- Svensson, P., & Saltzstein, B. (n.d.). Wireless technologies for medical devices: Choices and Trade-offs. Retrieved March 30, 2017, from <http://www.connectblue.com/press/articles/wireless-technologies-for-medical-devices-choices-and-trade-offs/>
- Termómetros: dicas para escolher e medir a temperatura. (2015). Retrieved December 15, 2016, from <https://www.deco.proteste.pt/saude/doencas/dicas/termometros-dicas-para-escolher-e-medir-a-temperatura>
- The Editors of Encyclopædia Britannica. (2005). Database. Encyclopædia Britannica. Retrieved from <https://www.britannica.com/technology/database>
- Thomas M. Connolly, C. E. B. (2004). *Database Solutions: A Step-by-step Guide to Building Databases* (2nd ed.). Harlow: Person Education Limited. Retrieved from https://books.google.pt/books?id=VsZO_kZl4bUC&printsec=frontcover&hl=pt-PT
- Tutorialspoint. (n.d.-a). Bootstrap Tutorial. Retrieved March 17, 2017, from https://www.tutorialspoint.com/bootstrap/bootstrap_overview.htm
- Tutorialspoint. (n.d.-b). jQuery Tutorial. Retrieved March 17, 2017, from

- <https://www.tutorialspoint.com/jquery/jquery-selectors.htm>
Tutorialspoint. (n.d.-c). Laravel - Overview. Retrieved March 17, 2017, from https://www.tutorialspoint.com/laravel/laravel_discussion.htm
- Tutorialspoint. (n.d.-d). UML - Activity Diagrams. Retrieved February 23, 2017, from https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
- Tutorialspoint. (n.d.-e). What is CSS? Retrieved March 17, 2017, from https://www.tutorialspoint.com/css/what_is_css.htm
- Usability Testing. (n.d.). Retrieved July 4, 2017, from [https://www.agilealliance.org/glossary/usability/#q=~\(filters~\(postType~\(~%27page~%27post~%27aa_book~%27aa_event_session~%27aa_experience_report~%27aa_glossary~%27aa_research_paper~%27aa_video\)~tags~\(~%27usability*20testing\)\)~searchTerm~%27~sort~false~sortDirection~%27asc~page~1\)](https://www.agilealliance.org/glossary/usability/#q=~(filters~(postType~(~%27page~%27post~%27aa_book~%27aa_event_session~%27aa_experience_report~%27aa_glossary~%27aa_research_paper~%27aa_video)~tags~(~%27usability*20testing))~searchTerm~%27~sort~false~sortDirection~%27asc~page~1))
- Utentes referenciados por centros de saúde ou Saúde 24 têm prioridade. (2016). Retrieved November 23, 2016, from <https://www.sns.gov.pt/noticias/2016/08/01/hospitais-atendimento-prioritario/>
- Vallejos de Schatz, C. H., Medeiros, H. P., Schneider, F. K., & Abatti, P. J. (2012). Wireless Medical Sensor Networks: Design Requirements and Enabling Technologies. *Telemedicine and E-Health*, 18(5), 394–399. <https://doi.org/10.1089/tmj.2011.0169>
- Van Haute, T., De Poorter, E., Crombez, P., Lemic, F., Handziski, V., Wirström, N., ... Moerman, I. (2016). Performance analysis of multiple Indoor Positioning Systems in a healthcare environment. *International Journal of Health Geographics*, 15(1), 7. <https://doi.org/10.1186/s12942-016-0034-z>
- Vasisht, D., Kumar, S., & Katabi, D. (2016). Decimeter-Level Localization with a Single WiFi Access Point. *Nsdi 2016*, 165–178. Retrieved from <https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-vasisht.pdf>
- Visão. (2015). Medicina de Guerra. Retrieved from <http://visao.sapo.pt/actualidade/sociedade/medicina-de-guerra=f808417>
- ViSi Mobile System. (n.d.). Retrieved January 12, 2017, from <http://www.soterawireless.com/visi-mobile/>
- What is Bluetooth? (n.d.). Retrieved March 28, 2017, from <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works>
- World Health Organization. (2011). Pulse Oximetry Training Manual. *World Health Organization*, 1–23.
- XML Soap. (n.d.). Retrieved June 14, 2017, from https://www.w3schools.com/xml/xml_soap.asp
- ZigBee® Wireless Standard. (n.d.). Retrieved January 24, 2017, from <https://www.digi.com/resources/standards-and-technologies/rfmodems/zigbee-wireless-standard>