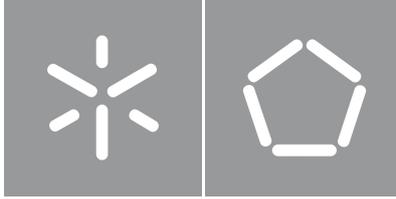


Universidade do Minho
Escola de Engenharia

Carlos João Teixeira Preto

**e-Saúde - Design e Desenvolvimento de
Serviços Web de Telemonitorização**



Universidade do Minho
Escola de Engenharia

Carlos João Teixeira Preto

**e-Saúde - Design e Desenvolvimento de
Serviços Web de Telemonitorização**

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação de
José Francisco Creissac Freitas de Campos

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.



CC BY-NC-ND

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais pelo apoio e disponibilidade demonstrados sempre que necessário, tentando sempre garantir que eu dispunha das melhores condições para concluir com sucesso a minha licenciatura e mestrado.

Aos meus amigos e namorada, agradecer por me terem acompanhado ao longo deste percurso académico e me motivarem para querer sempre alcançar mais.

Ao Prof. José Francisco Creissac queria expressar o meu obrigado, por ter aceite ser o meu orientador na Universidade do Minho e me ter orientado e ajudado sempre que necessário com conselhos construtivos.

À *Altice Labs* um enorme agradecimento por me terem acolhido neste projeto, especialmente à minha orientadora Telma Mota e aos elementos da equipa *SmartAL* - Bruno Miguel, Bruno Medeiros, Ricardo Machado e Jean Matias, que me foram acompanhando regularmente e esclarecendo todas as dúvidas que eu pudesse ter.

Por fim, um agradecimento ao programa GENIUS Investigação da Inova-Ria, pelo voto de confiança depositado em mim e pela forma generosa como me trataram e acolheram no projeto.

Declaração de Integridade

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Universidade do Minho, Braga, outubro 2023

Carlos João Teixeira Preto

Resumo

A área da *eHealth* tem ganho uma enorme importância nos últimos anos. Tal advém não só da progressiva evolução tecnológica que se tem verificado ao longo dos tempos, mas também da necessidade de melhorar a qualidade de vida das pessoas que necessitam de assistência médica permanente ou ocasional. Mais especificamente, a telemonitorização permite aos pacientes estarem mais envolvidos no seu processo de recuperação e agirem mais rapidamente quando surgem novos problemas ou complicações. Neste tipo de serviços, existem geralmente três entidades, o paciente, o cuidador e os dispositivos tecnológicos, que funcionam como facilitadores entre ambos. Atualmente, existem cada vez mais serviços de telemonitorização, mas, na maioria dos casos, estes serviços seguem uma abordagem genérica, que pode ser demasiado complexa para alguns utilizadores. Tal deve-se ao facto de incluírem por vezes funcionalidades desnecessárias para certos doentes, o que não lhes permite a melhor experiência de navegação e utilização. No âmbito da dissertação, desenvolveu-se uma aplicação *web* que permite aos utentes registar as suas medições através de dispositivos *Bluetooth*, e responder a questionários providenciados pelos seus médicos. Quer as medições, quer as respostas aos questionários poderão ser consultadas por parte dos médicos dos pacientes, permitindo assim, tal como pretendido, uma melhor monitorização e controlo do estado de saúde dos pacientes. Esta aplicação pretende ser mais simples que as restantes, utilizando serviços modulares, para que os pacientes possam escolher o que querem utilizar, de acordo com o seu perfil, sempre e quando precisarem. Cada funcionalidade principal da aplicação pode ser vista como um micro-serviço independente. Adicionalmente, o projeto segue as mais recentes abordagens tecnológicas e arquiteturas. Em termos de interface e usabilidade, procurou-se desenvolver uma aplicação atrativa e intuitiva. Por fim, é importante mencionar que o presente projeto se enquadra no âmbito de desenvolvimento da solução de telemonitorização da *Altice Labs*, denominada *SmartAL*, que permite recolher dados do paciente a partir de dispositivos inteligentes e questionários.

Palavras-chave Telemonitorização, eHealth, Micro-Serviço, SmartAL

Abstract

The *eHealth* area has gained an enormous importance in recent years. That comes not only from the progressive technological evolution that has been happening over time, but also from the need to improve the quality of life of people who need permanent or occasional medical assistance. Telemonitoring in particular allows patients to be more involved in their process of recovery and to quickly act when a new problem appears. In this types of services, there are usually three entities, the patient, the caretaker and the technological devices, that work as facilitators between both. In today's world, more and more telemonitoring services are beginning to appear, but most often, these services follow a generic approach, that can be too complex for some users, because it includes many unnecessary features, that do not allow an easy navigation and orientation flow through the apps. In this dissertation scope, the goal is to develop a web application that allow patients to register their measurements through Bluetooth devices and also answer questionnaires given by their doctors. Both the measurements and the questionnaires will be available to the patients doctors, that allow for a better control of their patients health, just as intended. This application is simpler, and uses modular services, so that the patients can choose what they want to control whenever they need to. Each functionality can be seen as an independent microservice. The project also follows the most recent technological and architectural approaches. In terms of interface and usability, the app aims to be attractive and intuitive. Finally, it is important to refer that the present project fits in the *SmartAL* scope, which is a telemonitoring solution developed in *Altice Labs*, that has features that allow the gathering and analysis of data, collected through the use of smart devices.

Keywords Telemonitoring, eHealth, Microservices, SmartAL

Conteúdo

1	Introdução	1
1.1	Contexto e Motivação	1
1.2	Soluções <i>eHealth</i> da <i>Altice Labs</i>	2
1.3	Principais Objetivos	3
1.4	Estrutura do Documento	4
2	Estado da arte	5
2.1	Enquadramento	5
2.2	Serviços de Telemonitorização	7
2.2.1	Estrutura dos Serviços de Telemonitorização	8
2.2.2	Principais Doenças	9
2.3	Dispositivos	11
2.4	Comunicação	14
2.5	Pontos Fortes e Limitações	15
2.6	Casos de Aplicação	16
2.6.1	Telemonit SNS 24	17
2.6.2	HopeCare	18
2.6.3	Homedoctor	19
2.6.4	Análise Comparativa das Aplicações	20
2.7	Solução SmartAL	20
2.8	Conclusão	22
3	Modelação e Arquitetura	24
3.1	Requisitos	24
3.1.1	Requisitos Funcionais	24
3.1.2	Requisitos Não Funcionais	26

3.2	Casos de Uso	26
3.2.1	Realizar Medição Clínica	27
3.2.2	Consultar medições de paciente	29
3.3	Arquitetura Funcional	29
3.4	Diagramas de Sequência	30
3.4.1	Realizar medição clínica	31
3.4.2	Consultar medições de paciente	32
3.5	Modelos de Dados	32
3.5.1	Entities	33
3.5.2	Notifications	34
3.5.3	Measurements	35
3.5.4	Shared Info	36
3.5.5	Questionnaires	37
3.6	Conclusão	39
4	Implementação do Sistema	40
4.1	Ferramentas e Tecnologias	40
4.2	Implementação	43
4.2.1	Serviços	43
4.2.2	Frontend	52
4.3	Interfaces	59
4.3.1	Paciente	61
4.3.2	Médico	66
4.4	Dispositivos Utilizados	69
4.5	Validação	70
4.6	Deployment	72
5	Conclusões e trabalho futuro	73
5.1	Conclusões	73
5.2	Perspetiva de trabalho futuro	75
	Bibliografia	76
A	Requisitos Funcionais	79

B	Requisitos Não Funcionais	84
C	Use Cases	86
C.1	Iniciar sessão	86
C.2	Editar thresholds	86
C.3	Consultar medições	87
C.4	Gerar PDF do histórico	88
C.5	Responder a questionário	88
C.6	Consultar questionário	89
C.7	Definir médicos de partilha	90
C.8	Editar thresholds de paciente	91
C.9	Consultar respostas de paciente a questionário	92
C.10	Realizar <i>upload</i> de questionário	93
C.11	Eliminar questionário	93
C.12	Consultar <i>template</i> de questionário	94
C.13	Ver perfil	94
D	Diagramas de Sequência	96
D.1	Editar thresholds	96
D.2	Consultar medições	97
D.3	Gerar PDF do histórico	97
D.4	Responder a questionário	98
D.5	Consultar questionário	99
D.6	Definir médicos de partilha	99
D.7	Editar thresholds de paciente	100
D.8	Consultar respostas de paciente a questionário	101
D.9	Realizar <i>upload</i> de questionário	101
D.10	Consultar <i>template</i> de questionário	102
D.11	Ver perfil	102

Lista de Figuras

1	Índice Envelhecimento Portugal	6
2	Estrutura Serviços Telemonitorização	8
3	Telemonit SNS 24	17
4	HopeCare	19
5	Homedoctor	19
6	Arquitetura SmartAL	21
7	Diagrama de Casos de Uso	27
8	Use Case - Realizar medição clínica	28
9	Use Case - Consultar medições de paciente	29
10	Arquitetura Funcional Final	30
11	Diagrama Sequência - Realizar medição clínica	31
12	Diagrama Sequência - Consultar medições de paciente	32
13	Modelo de Dados do Serviço <u>Entities</u>	33
14	Modelo de Dados do Serviço <u>Notifications</u>	34
15	Modelo de Dados do Serviço <u>Measurements</u>	35
16	Modelo de Dados do Serviço <u>Shared Info</u>	37
17	Modelo de Dados do Serviço <u>Questionnaires</u>	38
18	Ferramentas e Tecnologias	41
19	Fluxo de um Pedido	44
20	Implementação do Serviço <u>Measurements</u>	45
21	Estrutura <u>MeasurementController</u>	46
22	Endpoint para adicionar <u>Medição</u>	46
23	Implementação do Serviço <u>Questionnaires</u>	48
24	Estrutura <u>QuestionnaireController</u>	49

25	Endpoint para criar nova sessão	49
26	Implementação do Serviço Shared Info	51
27	Estrutura PatientDoctorController	51
28	Endpoint para verificar privilégios	52
29	Estrutura Frontend	53
30	Estrutura Components	54
31	Estrutura Services	55
32	Lista de Propriedades dos Dispositivos	56
33	Dispositivos Existentes	56
34	Função EventHandler	57
35	Função connect	58
36	Login	59
37	Editar Informação	60
38	Notificações	60
39	Feedback	61
40	Página das Medições	61
41	Página com Histórico	62
42	Página de Gerar PDF	63
43	Relatorio Gerado	64
44	Os Meus Médicos	64
45	Questionários	65
46	Responder Questionário	66
47	Perfil Paciente	66
48	Os Meus Pacientes	67
49	Detalhes de um Paciente	68
50	Questionários	68
51	Perfil Médico	69
52	Dispositivos	70
53	Validação Requisitos Funcionais	71
54	Validação Requisitos Não Funcionais	71
55	Use Case - Iniciar Sessão	86
56	Use Case - Editar <i>thresholds</i>	87

57	Use Case - Consultar medições	87
58	Use Case - Gerar PDF do histórico	88
59	Use Case - Responder a questionário	89
60	Use Case - Consultar questionário	90
61	Use Case - Definir médicos de partilha	91
62	Use Case - Editar thresholds de paciente	92
63	Use Case - Consultar questionário	92
64	Use Case - Realizar upload de questionário	93
65	Use Case - Eliminar questionário	94
66	Use Case - Consultar template de questionário	94
67	Use Case - Ver perfil	95
68	Diagrama Sequência - Editar thresholds	96
69	Diagrama Sequência - Consultar medições	97
70	Diagrama Sequência - Gerar PDF do histórico	97
71	Diagrama Sequência - Responder a questionário	98
72	Diagrama Sequência - Consultar questionário	99
73	Diagrama Sequência - Definir médicos de partilha	99
74	Diagrama Sequência - Editar thresholds de paciente	100
75	Diagrama Sequência - Consultar respostas de paciente a questionário	101
76	Diagrama Sequência - Realizar upload de questionário	101
77	Diagrama Sequência - Consultar template de questionário	102
78	Diagrama Sequência - Ver perfil	102

Acrónimos

API *Application Programming Interface.*

B2B Business to Business.

B2C Business to Consumer.

DPOC Doenças Pulmonares Obstrutivas Crônicas.

ECA Enzima Conversora de Angiotensina.

ECG Electrocardiograma.

INE Instituto Nacional de Estatística.

IoT *Internet of Things.*

MAPA Medição em Ambulatório da Pressão Arterial.

OMS Organização Mundial da Saúde.

SNS Serviço Nacional de Saúde.

Capítulo 1

Introdução

1.1 Contexto e Motivação

A progressiva evolução tecnológica tem tido um enorme impacto, não só na sociedade no seu conjunto, como também individualmente, em cada um de nós. Através do recurso à tecnologia, muitas das típicas tarefas diárias, e em particular as relacionadas com a nossa saúde, como por exemplo as hospitalares, tais como execução exames clínicos, diagnósticos, interpretação de dados e tratamento, podem agora ser realizadas, não só com mais facilidade, como também com maior eficácia e rapidez.

Na área da saúde, tal evolução tecnológica fez com que surgissem novas ramificações dentro da própria área, sendo uma delas a *eHealth*. Mas o que é de facto *eHealth*, e como se caracteriza? *eHealth* [19], ou medicina digital, representa, de forma abrangente, os recursos tecnológicos utilizados na área da saúde que contribuem para melhorar o auxílio prestado aos doentes, tendo como principal objetivo aumentar a eficiência, eficácia e acessibilidade aos cuidados de saúde.

Dentro desta área, uma das vertentes que tem ganho cada vez mais relevância é a da telesaúde, que consiste no acompanhamento remoto de pacientes, através de meios tecnológicos de comunicação e sensorização. Especificamente, a telemonitorização surgiu com o intuito de melhorar a qualidade de vida das pessoas que necessitam de acompanhamento regular, seja porque apresentam doenças crónicas, porque obtiveram alta hospitalar recentemente e necessitam de acompanhamento regular numa fase inicial do seu período de recuperação, ou porque simplesmente se encontram em qualquer outro estado de fragilidade temporária ou permanente que necessita de atenção. O acompanhamento à distância permite envolver mais as pessoas no seu processo de recuperação, agir com mais rapidez em caso de problemas e gerir com mais eficiência os recursos institucionais e humanos.

Neste tipo de serviços existem tipicamente três entidades envolvidas, o paciente, o cuidador, e os dispositivos tecnológicos de recolha de informação, que funcionam como facilitadores entre ambos. A forma como a informação é recolhida pode ter como base dispositivos clínicos sensorizados, *wearables*

ou qualquer outra fonte de suporte digital ou manual, como por exemplo ficheiros, imagens ou questionários. No caso específico da recolha de sinais vitais, o paciente recorre normalmente a dispositivos clínicos, predominantemente *Bluetooth*, que lhe permite realizar as medições necessárias, e posteriormente aceder à informação através de uma aplicação móvel ou *web*. Tais registos podem ser partilhados pelo paciente à posteriori com o seu cuidador ou disponibilizados em tempo real pelas próprias plataformas de telemonitorização. Neste caso, o cuidador poderá tomar decisões e agir de imediato em casos de risco com base nos dados recebidos.

Através do recurso à telemonitorização, os pacientes controlam de forma mais regular a sua saúde, evitam deslocações sucessivas aos centros médicos e reduzem as situações de *stress* associadas a tais deslocações. Porém, as vantagens não são exclusivas dos pacientes. Este tipo de serviço contribui para uma gestão mais eficiente do tempo dos profissionais de saúde e para o alívio da excessiva carga de doentes nas instituições de saúde, visto que os pacientes não terão necessidade de se deslocar tão frequentemente aos centros de saúde e/ou hospitais.

Não obstante, verifica-se que a maioria das aplicações associadas a este tipo de serviço adotam uma abordagem demasiado genérica, e, por vezes, complexa para determinados pacientes, apresentando muitas vezes funcionalidades desnecessárias, ou demasiado específicas (dependente de determinados dispositivos), não permitindo, de forma ágil, acrescentar outras funcionalidades em casos de comorbidades. Por exemplo, um paciente com diabetes, caso adira a um serviço de telemonitorização, poderá não ter acesso a um serviço personalizado de controlo da alimentação, que é essencial neste tipo de doença, mas terá acesso a funcionalidades relacionadas com outras doenças, tais como as relacionadas com problemas respiratórios e cardíacos, que não precisa de momento. Esta complexidade e falta de personalização pode complicar a utilização das aplicações. Neste contexto, é necessário construir aplicações mais simples, utilizando serviços modulares, para que assim os pacientes tenham acesso apenas ao que necessitem e também possam alterar a informação pretendida sempre e só quando for necessário.

1.2 Soluções eHealth da Altice Labs

Fundada em 1950, a *Altice Labs* (anteriormente Centro de Estudos de Telecomunicações e posteriormente PT Inovação), é a empresa tecnológica de inovação do grupo *Altice*, que se foca no desenvolvimento de produtos e serviços inovadores para o mercado das telecomunicações e das tecnologias de informação. Os produtos desenvolvidos na *Altice Labs* têm como objetivo melhorar a qualidade de vida das pessoas e contribuir para uma progressiva evolução tecnológica na sociedade [12]. Dos vários valores presentes

na empresa, destacam-se a criatividade, inovação, dedicação, espírito de equipa, capacidade de aprendizagem e excelência.

Adicionalmente, a *Altice Labs* promove a cooperação com as Universidades e outros institutos de investigação e desenvolvimento, sejam eles nacionais ou internacionais, assumindo-se assim, não apenas como um verdadeiro agente da transferência do conhecimento para o mercado e para a indústria, como também um motor e agente de transformação e inovação tecnológica da sociedade.

Na área *eHealth*, a empresa desenvolveu dois produtos, o *Medigraf* [13], uma solução de comunicação e colaboração criada para os profissionais de saúde das mais diversas áreas, tendo como foco a realização de teleconsultas e partilha em *streaming* de informação proveniente de dispositivos médicos, e o *SmartAL*. O *SmartAL* [14] corresponde a uma solução de telemonitorização capaz de recolher dados de diversos dispositivos clínicos e não clínicos, disponibilizando também outros mecanismos de recolha de informação sobre pacientes, como por exemplo questionários, imagens e ficheiros.

1.3 Principais Objetivos

O presente projeto enquadra-se no contexto do produto *SmartAL*, tendo como foco desenvolver um sistema modular de telemonitorização, composto por serviços simples e independentes que se adequem às necessidades de cada paciente em cada momento, ao invés de ter uma aplicação monolítica e genérica para todos os pacientes. Deve também seguir as novas abordagens tecnológicas, para se tornar mais atrativa, quer para pacientes, quer para profissionais de saúde.

Cada funcionalidade de telemonitorização poderá ser vista como um micro-serviço independente, onde cada paciente, com base na sua doença, poderá escolher as que melhor lhe convier, ou ter acesso, na interface da aplicação, apenas à informação necessária de acordo com o seu perfil. Em qualquer um dos casos, o utilizador deverá conseguir visualizar, através da aplicação, a informação recolhida pelas medições realizadas, aceder ao seu histórico, bem como enviar a informação que achar conveniente ao seu cuidador, sempre e quando desejar.

No contexto específico da dissertação, irão ser utilizados alguns dispositivos *Bluetooth* que permitam a validação da aplicação desenvolvida, de modo a que no futuro, quando for necessário adicionar compatibilidade com aparelhos que efetuem outro tipo de medições, apenas seja necessário analisar os dados obtidos por esse mesmo aparelho e proceder ao seu tratamento, não sendo necessário efetuar nenhuma alteração no código previamente desenvolvido.

1.4 Estrutura do Documento

Neste primeiro capítulo, foi apresentado o contexto em que a atual dissertação se insere, os principais motivos e objetivos que desencadearam a realização do projeto e uma breve apresentação das soluções *eHealth* desenvolvidas pela empresa responsável pela proposta, a *Altice Labs*.

No segundo capítulo, é explorado o estado atual da área associada ao projeto, desde o enquadramento e situação atual da telemedicina, até à descrição do funcionamento geral dos serviços de telemonitorização, assim como dos dispositivos mais utilizados neste tipo de serviços. Para além disso, são ainda apresentados alguns exemplos existentes no mercado, realçando os seus aspetos mais inovadores e as suas desvantagens. Estes servirão como referência na concepção dos novos serviços de telemonitorização. Por fim, é abordado o estado da prática associado ao produto *SmartAL*, e como este será importante para o desenvolvimento do projeto.

No terceiro capítulo aborda-se a modelação e arquitetura do sistema, desde os requisitos funcionais e não-funcionais levantados, os casos de uso definidos e a devida explicação de cada um deles, a arquitetura funcional, o modelo de dados de cada serviço desenvolvido e, por fim, os diagramas de sequência associados a cada um dos casos de uso implementados.

No quarto capítulo são exploradas as tecnologias utilizadas e todo o processo de implementação da aplicação, desde a *backend* até à *frontend*. Para complementar, são ilustradas as interfaces da aplicação final, os dispositivos *Bluetooth* utilizados e o processo de validação e de *deployment*.

Por fim, no quinto e último capítulo encontram-se quer a conclusão, quer o trabalho futuro a ser realizado.

Capítulo 2

Estado da arte

2.1 Enquadramento

Em 1947, a Organização Mundial da Saúde definiu o termo saúde como "...um estado de completo bem-estar físico, mental e social, e não apenas a ausência de doença"[5]. Apesar de ser uma definição bastante avançada para a época em que foi proferida, percebe-se que, de facto, a maioria das pessoas sempre considerou o conceito de doença como o principal fator diferenciador na percepção da saúde, e tal deve-se ao impacto negativo que esta pode ter no quotidiano das pessoas. Segundo o jornal Expresso [6], um estudo realizado pelo Instituto Nacional de Estatística em 2021 concluiu que 43,9% da população residente em Portugal com idade superior a 16 anos apresentou prevalência de doenças crónicas ou problemas de saúde. Estes valores são os mais elevados desde 2016, correspondendo a um aumento de 0.7 pontos percentuais relativamente aos valores obtidos em 2020, e 2.7 pontos percentuais acima dos valores obtidos em 2019.

Mas o que são de facto doenças crónicas? A **OMS** descreve as doenças crónicas como [3] doenças permanentes, que produzem incapacidades das mais diferentes formas nos doentes, sendo causadas por alterações patológicas irreversíveis. Dentro das doenças crónicas, é ainda importante fazer a devida distinção entre as doenças crónicas congénitas, doenças determinadas pela genética, e as doenças crónicas não congénitas, que correspondem àquelas que se desenvolvem ao longo da vida, e em que os tratamentos podem ou não levar à cura total do paciente. Estas últimas são as mais preocupantes e as que têm vindo a aumentar ao longo dos anos, principalmente em pessoas idosas, como seria expectável. À medida que os anos passam, o nosso estado de saúde vai ficando cada vez mais vulnerável, não tendo o nosso corpo as forças nem os recursos necessários para superar os diversos obstáculos que a vida nos coloca. Este aumento de vulnerabilidade é propício ao aparecimento e desenvolvimento de vários problemas de saúde, entre eles, as doenças crónicas. Segundo as Nações Unidas [21], estima-se que o número de idosos com 60 anos ou mais duplique até 2050 e mais do que triplique até 2100, passando

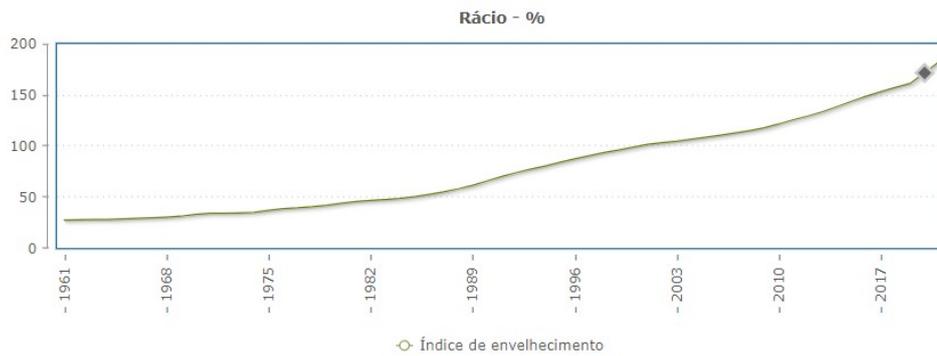


Figura 1: Índice Envelhecimento Portugal

de 962 milhões em 2017 para 2,1 mil milhões em 2050 e 3,1 mil milhões em 2100. De facto, os valores apresentados são bastante expressivos, e requerem que se comece a pensar seriamente em novas soluções/estratégias para, no mínimo, ajudar a mitigar o impacto que o envelhecimento da população tem no próprio indivíduo e na sociedade.

E como é que esses números se traduzem em Portugal? Um estudo realizado pela Pordata e divulgado pela Sic Notícias [17] confirmou que Portugal é o país da União Europeia que está a envelhecer mais rapidamente. Como se pode observar na Figura 1, o rácio de idosos por cada 100 jovens tem vindo a aumentar, tendo Portugal um valor atual de cerca de 182 idosos por cada 100 jovens, números que contrastam com valores anteriores, como os de 1990, onde havia um rácio de 66 idosos por cada 100 jovens.

Apesar dos valores apresentados serem bastante significativos, há um dado que torna a situação ainda mais preocupante. Analisando dados divulgados pelo Jornal Público [20], num estudo realizado em cinco países europeus, Portugal é o país que apresenta o pior resultado no envelhecimento saudável. No mesmo documento, faz-se ainda referência a algumas potenciais razões para tais valores, sendo importante destacar a falta de conhecimento por parte da população idosa, das novas tecnologias e novos métodos de tratamento - a chamada iliteracia digital na área da saúde. De facto, apesar de nos últimos anos ter havido um enorme progresso na medicina, existe uma porção significativa da população que desconhece estes novos métodos, tais como por exemplo os usados na telemonitorização, o que faz com que o estado de saúde destas pessoas não possa ser regulado da melhor forma.

Temos então que o grande sucesso da medicina, que se traduz no aumento da esperança de vida e no declínio da mortalidade infantil, tem impacto direto no envelhecimento geral da população e no decorrente aumento das doenças crónicas, sobretudo nos países mais desenvolvidos. Neste contexto, é necessário arranjar mecanismos que mitiguem o impacto negativo desta situação, para que assim se

possa providenciar uma qualidade de vida superior aos doentes.

Uma das soluções mais adotadas consiste em acompanhar estes doentes de forma regular, incentivando-os a deslocarem-se aos centros médicos sempre que seja necessário realizar a monitorização ativa dos diferentes parâmetros relacionados com a saúde e o bem-estar. Porém, muitos dos pacientes já não apresentam as melhores capacidades motoras, e muitas vezes habitam em zonas rurais, de acessos complicados, tornando-se bastante difícil realizar viagens frequentes aos centros médicos. Adicionalmente, desde que surgiu a COVID-19, tornou-se cada vez mais difícil agendar consultas. Segundo um estudo realizado pelo **INE** [6], em 2021, 5.7% da população não conseguiu agendar uma consulta médica em tempo útil, acrescentando ainda que os valores têm vindo progressivamente a aumentar desde 2019.

Uma outra solução frequentemente utilizada é o recurso a lares ou instituições de acolhimento, pois tanto os próprios como a família deixam de ter condições para cuidar dos idosos. O problema é que, à medida que as pessoas envelhecem, vão-se tornando cada vez mais sensíveis ao meio em que vivem, e quando estas são colocadas em lares, é necessário um longo período de adaptação ao novo meio, onde existem novas rotinas e espaços. A tendência é de que com a idade, a capacidade adaptativa das pessoas diminua, o que faz com que nestes casos, a longa demora na adaptação possa levar a problemas psicológicos diversos, tais como a depressão. Deste modo, regra geral, os lares apenas deverão ser vistos como último recurso.

Todos estes fatores fazem com que seja necessário adequar e desenvolver novos métodos capazes de ajudar a mitigar os problemas inerentes ao envelhecimento e ao aparecimento de doenças crónicas. Neste contexto, a tecnologia e a utilização de meios digitais que caracterizam a *eHealth* têm vindo a ganhar destaque nos últimos anos. Em particular, no âmbito da tele saúde, o acompanhamento remoto de doentes que se traduz na prestação de serviços de teleconsulta e telemonitorização clínica, constituem duas das ferramentas principais na melhoria da prestação de cuidados. O foco deste trabalho será na telemonitorização.

2.2 Serviços de Telemonitorização

A telemonitorização consiste na utilização de tecnologia e de meios de comunicação adequados para providenciar cuidados de saúde e suporte à distância a pacientes nas suas casas. Recorrendo a diversos dispositivos inteligentes, mais conhecidos como dispositivos *Internet of Things (IoT)*, é possível capturar dados em tempo real. Tipicamente, no ato presencial da consulta, os pacientes recebem um ou mais dispositivos de medição para verificarem regularmente os seus sinais vitais, e no caso de haver algo de

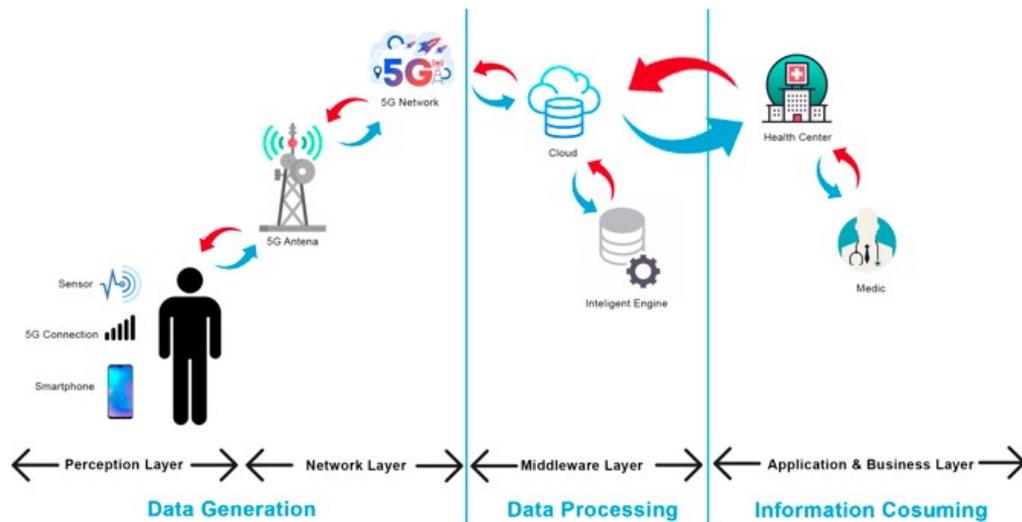


Figura 2: Estrutura Serviços Telemonitorização

errado, os profissionais de saúde responsáveis pelo paciente são notificados e posteriormente decidem como agir. Esta coleta de informação de forma preferencialmente automática permite aos pacientes estarem em constante monitorização, tornando-se muito mais fácil detetar o possível agravamento do seu estado de saúde ou surgimento de um qualquer problema novo. Adicionalmente, em muitos casos, os resultados provenientes dos pacientes são utilizados como dados de entrada em modelos inteligentes, para ajudar no reconhecimento de novos padrões, permitindo assim apostar na prevenção e detecção mais rápida de possíveis sinais de risco.

No entanto, apesar de todas as reconhecidas vantagens da telemonitorização, esta não deve substituir as típicas visitas médicas, mas sim funcionar como complemento e auxílio, de forma a gerir melhor o tempo dispendido na assistência aos doentes e reduzir o risco de hospitalização.

2.2.1 Estrutura dos Serviços de Telemonitorização

Os serviços de telemonitorização compreendem várias etapas, algumas delas complexas. Para construir um serviço adequado às necessidades de pacientes e profissionais, é necessário pensar cuidadosamente em cada uma delas e especificar bem os seus requisitos. Para melhor entender o funcionamento de um serviço típico de telemonitorização, é necessário analisar a sua estrutura, desde a aquisição e partilha de dados, até à apresentação dos mesmos aos utilizadores [7].

Na Figura 2 encontra-se representada a arquitetura de referência deste tipo de serviços. Normalmente, os sistemas são constituídos por 4 camadas. A primeira, e possivelmente aquela com a qual

o paciente tem mais contacto, é a camada *Perception*. Esta camada envolve todos os aparelhos e tecnologias associados à realização de diferentes leituras e medições, tais como telemóveis, dispositivos *Bluetooth*, sensores.

Uma vez recolhidos e preparados, os dados são enviados para uma outra camada, a camada *Network*, responsável por conectar todos os dispositivos. Os dados são partilhados, geralmente através de *Bluetooth*, ou, em alguns casos, através *WiFi* ou rede móvel. A terceira camada da arquitetura denomina-se *Middleware*, e é constituída pelos diferentes serviços necessários para tratar e armazenar numa base de dados a informação recolhida na camada *Network*.

Por fim, a última camada, *Application/Business*, compreende serviços de saúde que integram os dados obtidos pelas camadas anteriores, sendo por isso a principal responsável por controlar todas as funcionalidades oferecidas ao utilizador e apresentar a informação corretamente, recorrendo para isso a tabelas, gráficos, diagramas e outros modelos.

Cada camada realiza a sua própria função, porém, é ainda possível agrupar camadas com base nos processos que as envolvem, de forma a diferenciar as diferentes etapas presentes nos serviços. As camadas *Perception* e *Network* correspondem à etapa *Data Generation*, referente ao processo de recolha e geração de dados pelos diversos dispositivos. Por sua vez, a camada *Middleware* está associada à etapa de *Data Processing*, visto que engloba a análise dos dados recolhidos, geralmente através de algoritmos de *machine learning*. Relativamente à camada *Application/Business*, é possível incluí-la na etapa de *Data Consuming*, visto que o resultado final do processo irá ser consumido e visualizado, tanto pelos doentes como pelos profissionais de saúde, sendo que estes últimos tomarão decisões com base nos dados recolhidos.

2.2.2 Principais Doenças

Após entendida a estrutura deste tipo de serviços, é importante saber quais as doenças em que estes estão a ser mais utilizados e a ter mais sucesso. Atualmente, os serviços de telemonitorização são maioritariamente utilizados por pacientes com problemas de saúde associados a três áreas distintas, sendo estas a cardiologia, a pneumologia e a endocrinologia [15]. Estas são também as áreas da medicina que acompanham algumas das doenças crónicas mais prevalentes.

Dentro da cardiologia, estes serviços são bastante utilizados no acompanhamento de doentes com hipertensão. A elevada pressão arterial é um problema cada vez mais comum e que tende a agravar-se com a idade. A hipertensão é caracterizada pela pressão excessiva nas paredes das artérias e, segundo o SNS [4], afeta quase 40% da população europeia, e 42,6% da população adulta em Portugal. Os pacientes

que sofrem de hipertensão são, geralmente, assintomáticos. Isto é, não apresentam qualquer sintoma que possa levar à deteção da doença até ser demasiado tarde, e como tal, a única maneira de identificar corretamente se um paciente apresenta elevada pressão arterial é através de medições regulares.

Recorrendo aos dispositivos de monitorização, é possível efetuar uma melhor gestão da medicação dos pacientes com hipertensão. O tratamento destes pacientes é contínuo e requer várias medições, as quais se refletem no ajuste da medicação que melhor se adequa a cada situação. Apesar de existirem diversos tipos de medicamentos, a maioria está incluída numa de 10 classes distintas [2], sendo algumas das mais conhecidas as dos diuréticos, betabloqueadores, bloqueadores do canal de cálcio, vasodilatadores, inibidores da enzima conversora da angiotensina (**ECA**) e modificadores adrenérgicos. A única maneira de saber aquela que melhor se ajusta a um determinado paciente irá depender de fatores como o sexo, idade, etnia e problemas de saúde, entre outros. Na maioria dos casos, a combinação de dois ou mais de medicamentos é mais eficiente, visto que permite regular melhor a pressão sanguínea [8]. Com base nos valores da pressão arterial, é muitas vezes necessário ajustar a medicação, bem como a dose.

Ainda dentro da cardiologia, é importante analisar as alterações de peso nos pacientes, permitindo assim precaver cenários de insuficiências cardíacas. Um ganho súbito no peso diário de um paciente pode ser indicador da retenção de fluídos no seu corpo. Tal retenção pode ocorrer em diferentes partes do corpo, tais como, pulmões, rins, abdômen, pernas e pés. Recorrendo a registos manuais ou, preferencialmente, usando balanças inteligentes que permitam o registo automático de medições, os cardiologistas podem receber alertas quando a descompensação cardíaca ocorre, permitindo assim um tempo de reação mais rápido, que muitas vezes pode ser crucial no atendimento ao doente.

Contudo, o aumento de peso não está apenas associado ao ganho de fluídos. A obesidade é um dos maiores problemas de saúde na sociedade, podendo estar na origem de outras doenças graves, tais como a diabetes, o cancro e a elevada pressão arterial. No relatório de 2022 da **OMS** sobre obesidade na região Europeia [20], divulgou-se que, em Portugal, 57,5% da população tem excesso de peso, sendo que desses, 20,8% são obesos. Relativamente às crianças, cerca de uma em cada três crianças tem excesso de peso e mais de 10% delas têm mesmo obesidade. Neste contexto, a utilização de balanças, de preferência, inteligentes, permite aos cardiologistas fornecer *feedback* aos seus pacientes, aconselhando novas rotinas e dietas.

Relativamente à área da pneumologia, a telemonitorização permite uma melhor gestão das doenças respiratórias, e em particular as doenças pulmonares obstrutivas crónicas (**DPOC**). Apesar de serem praticamente incuráveis, é possível controlar e mitigar alguns dos seus impactos negativos. Para tal, os pneumologistas trabalham diretamente com os pacientes, de forma a desenvolver planos personalizados,

que combinam medicação, terapias de oxigênio e programas de reabilitação e suporte. Dispositivos como oxímetros e espirômetros ajudam a medir e controlar o estado da doença. Em doenças como a asma por exemplo, a utilização de monitores respiratórios eletrônicos permite aos pneumologistas definir planos para os seus pacientes, de forma a controlar complicações asmáticas. Tais planos envolvem exercício físico controlado e gestão de medicação.

Por fim, na área da endocrinologia, o principal foco dos serviços de telemonitorização encontra-se na gestão da glicose. A diabetes é uma doença crônica e incurável, porém, é possível diminuir o seu impacto na vida do paciente, através do controlo do peso, da alimentação e dos estilos de vida que se querem mais proativos e dinâmicos. Para além disso, tem de se analisar com frequência o nível de açúcar no sangue. Para alguns pacientes, tal controlo tem de ser realizado de forma periódica, podendo até mesmo ser realizado múltiplas vezes ao dia, recorrendo a dispositivos medidores de glicose no sangue. Para pacientes com diabetes do tipo 1, são aconselháveis medições constantes do nível de açúcar, para que assim seja possível, não só detetar alterações em tempo próximo do real, como também variações abruptas e picos, pormenores que são impossíveis de se detetar nos típicos testes de dedo. Tais medições são possíveis recorrendo a sistemas de monitorização de glicose que contêm um pequeno sensor que se coloca normalmente no braço durante um determinados períodos de tempo.

2.3 Dispositivos

A utilização de dispositivos de telemonitorização traz uma nova dinâmica para os serviços de saúde. Tipicamente, um doente teria sempre de se deslocar a um centro de saúde e esperar que um profissional de saúde o atendesse e efetuasse as medições pretendidas. Neste novo contexto, o processo de utilização dos dispositivos passa a ficar ao encargo dos pacientes, ou eventualmente dos seus cuidadores informais, que nas suas casas conseguem realizar as medições necessárias. Esta nova responsabilidade do paciente faz com que ele compreenda melhor o seu estado de saúde, e, com o passar do tempo, também consiga autoavaliar-se, tendo em conta os padrões que foi observando ao longo do tempo, contribuindo assim para resultados mais positivos. Adicionalmente, concede-lhe mais autonomia e um envolvimento direto no seu processo de tratamento.

Apesar de existirem no mercado diversos dispositivos de monitorização remota, há alguns que se destacam, não só pela sua portabilidade, como também pelo nível de prevalência de certas doenças [16].

Monitor de Pressão Arterial: Este instrumento calcula a pulsação e o fluxo sanguíneo através da medição das alterações no movimento das artérias. É bastante utilizado em problemas de saúde tais

como a hipertensão, diabetes e falha nos rins, porém é recomendado para qualquer situação onde haja problemas de elevada pressão arterial. A medição frequente da pressão permite um controlo mais eficaz de possíveis problemas cardíacos, e ajuda, em muitos cenários, a obter medições mais realistas, visto que quando as medições são realizadas por um médico, geralmente, os pacientes ficam mais nervosos que o habitual, e por isso apresentam valores diferentes dos corretos e mais comuns. Este efeito é conhecido como o efeito da bata-branca, e nestes casos, o **MAPA**¹ (Medição em Ambulatório da Pressão Arterial) tira as dúvidas existentes, uma vez que mede a tensão arterial em 24 horas.

Glicómetro: O glicómetro possui uma tira de testes que, quando entra em contacto com uma gota de sangue, permite realizar a leitura da glicose. Este aparelho é bastante utilizado por pacientes com diabetes do tipo 1 e 2, sendo por vezes necessário recorrer à utilização do glicómetro várias vezes ao dia, para entender qual o impacto na alteração do valor de açúcar no sangue de fatores como medicação, alimentação ou stress. Atualmente, a maioria dos doentes insulino-dependentes usam dispositivos de monitorização contínua e em tempo perto do real de glicose, pois estes permitem ter uma ideia mais precisa da sua variação. Bombas de perfusão subcutânea como as *FreeStyle Libre*² ou *Medtronic*³, são comparticipadas a 100% pelo **SNS**. Segundo dados da Direção Geral de Saúde [18], mais de 4100 pessoas estão em tratamento com estes dispositivos, sendo quase metade crianças.

Oxímetro de Pulso: Este dispositivo médico consiste num *clip* discreto e não invasivo, onde se encontra o sensor, que deve ser colocado no dedo, ou no lóbulo da orelha do paciente, e é capaz de medir a saturação, ou seja, a quantidade de oxigénio que está a circular nas células vermelhas do sangue, através da análise de comprimentos de onda. Adicionalmente, o aparelho regista normalmente a pulsação do paciente.

Tipicamente, o aparelho é utilizado por pacientes com doenças respiratórias crónicas e problemas temporários relacionados com o aparelho respiratório, tais como pneumonias, pleurisias, asma e mais recentemente, viroses como a COVID-19. Nos pacientes com COVID-19, níveis de oxigénio baixos nas células do sangue são um forte indicador de um diagnóstico positivo e de possíveis complicações adicionais mais sérias.

ECG + Estetoscópio: Dispositivos médicos que permitem avaliar o funcionamento e batimento do coração, e efetuar eletrocardiogramas (**ECG**). São utilizados regularmente em exames de rotina e em

¹ <https://www.saudebemestar.pt/pt/medicina/cardiologia/mapa/>

² <https://www.freestyle.abbott/pt-pt/home.html>

³ <https://www.medtronic-diabetes.com/pt-PT>

pacientes com problemas cardíacos, tais como arritmias e doenças arteriais. O estetoscópio amplifica os sons internos do corpo humano, sendo por isso capaz de capturar os sons do batimento cardíaco, pulmões e intestinos. Atualmente, é cada vez mais comum a utilização destes dois aparelhos, principalmente em pacientes em telemonitorização. Ao conjugar os resultados obtidos, os profissionais de saúde conseguem detetar possíveis anomalias nos registos cardíacos dos pacientes, e assim prevenir a propagação de futuras doenças. É também possível, utilizando dispositivos mais recentes, efetuar a medição contínua do **ECG**, como por exemplo o *S-Patch*⁴. Estes aparelhos são reutilizáveis e podem monitorizar o doente até 3-4 dias.

Termómetro: O termómetro é um aparelho bastante comum, estando presente na maioria das casas de hoje em dia e tendo como função fornecer indicação acerca da temperatura corporal da pessoa. São dispositivos clínicos e podem ser de diversos tipos, sendo alguns *Bluetooth*. Os digitais *non-touch* medem a temperatura aproximando o sensor da área da testa ou do ouvido, e os mais usuais, os termómetros de contacto, podem ser colocados por debaixo do braço do paciente ou por baixo da língua.

Balança: Outro instrumento muito comum e utilizado é a balança, responsável por medir o peso de uma pessoa. Apesar de ser recomendado controlar o peso, qualquer que seja a idade ou sexo do indivíduo, é essencial que pessoas em regime de telemonitorização com problemas de obesidade consultem regularmente o seu peso. Deste modo, os profissionais de saúde podem agir em conformidade, adaptando as dietas e os planos de exercício, prevenindo assim o risco de ataques cardíacos e de doenças perigosas, como a diabetes.

Wearable (Rastreador de Atividade e Monitorização Contínua): Atualmente, os *smart devices*, equipamentos como relógios e pulseiras inteligentes, são utilizados para controlar a atividade física e o bem-estar geral. Estes *wearables*, para além de serem bastante discretos, fornecem um leque extra de funcionalidades importantes que permitem rastrear e monitorizar cada indivíduo, efetuando várias medições, tais como passos diários, pulsação, risco de queda e qualidade do sono. Adicionalmente, a maioria destes equipamentos é altamente personalizável, levando a que muitas pessoas adquiram a este tipo de dispositivos a título pessoal. No entanto, dada a sua massificação, muitos são também usados como complemento à telemonitorização clínica, sendo que alguns destes *wearables* estão já certificados como dispositivos médicos. Normalmente, a sua precisão não se compara à dos dispositivos clínicos,

⁴ <https://www.samsungsds.com/eu/cardio/cardio.html>

mas podem sempre dar uma grande ajuda, tanto na recolha de informação complementar, como na confirmação de alguns valores, dada a sua monitorização contínua.

2.4 Comunicação

Mas como é que todos estes aparelhos comunicam? Primeiramente, é necessário entender como é que, através dos nossos telemóveis, é possível obter os valores medidos pelos dispositivos utilizados pelos pacientes. Todos os equipamentos acima mencionados contêm sensores capazes de recolher os mais diversos parâmetros e valores. Na maioria dos dispositivos clínicos, a informação é enviada por *Bluetooth* para o *smartphone* dos pacientes, sendo posteriormente reencaminhada para armazenamento em *cloud*.

O *Bluetooth* é uma tecnologia responsável por conectar dois ou mais aparelhos de forma remota, sem necessidade de recorrer a meios físicos, e permitir que troquem informação entre si. Para que tal comunicação possa ser estabelecida, ambos os aparelhos têm de ter adaptadores *Bluetooth*, que integram a capacidade de codificar e decodificar os dados através de ondas de rádio.

Os dispositivos *Bluetooth* podem ser de dois tipos, *master* ou *slave*, sendo que cada um dos tipos tem comportamentos distintos. O dispositivo *master* é responsável por iniciar a comunicação com os *slaves*, enviando pedidos a todos os aparelhos que se encontrem dentro de uma certa distância, que tipicamente ronda os 10 metros. Os *slaves* que receberem os pedidos do *master* respondem com o seu *Bluetooth Device Address*, que corresponde a um indicador único de 48-bit.

Na etapa seguinte, o dispositivo *master* irá selecionar o endereço que pretende e começa o processo de sincronização com o dispositivo *slave*. Este processo denomina-se *Service Discovery* e geralmente apenas necessita de ocorrer uma vez, pois assim que o *master* descobre o endereço do *slave*, regista-o. Uma vez sincronizados, uma *piconet*⁵ é estabelecida, e, os aparelhos que nela se encontrem, podem começar a trocar pacotes entre si.

No caso da telemonitorização clínica, o processo é bastante semelhante. Primeiro, o paciente liga o modo *Bluetooth* no seu telemóvel, que será o dispositivo *master*, e no seu dispositivo médico, que será o dispositivo *slave*. Neste tipo de dispositivos, o *Bluetooth*, quando existe, ativa automaticamente quando se liga o dispositivo. Posteriormente, o *smartphone* irá procurar dispositivos *Bluetooth* e quando encontrar o dispositivo médico, o paciente poderá verificar, na sua aplicação, que o mesmo foi reconhecido, e emparelhar os dois aparelhos, geralmente através de um botão de emparelhamento que aparece no ecrã do *smartphone*. Concluídos estes passos, os dois aparelhos podem então começar a comunicar.

⁵ *Piconet* representa uma micro-rede que conecta aparelhos via *Bluetooth*

Os dispositivos podem ainda comunicar com a *cloud*, através do telemóvel, por *WiFi* e/ou rede móvel, disponibilizando a informação a aplicações móveis ou *web*, através de **API**'s apropriadas. Estas permitem que as aplicações recolham os seus dados de forma fácil e segura, utilizando para isso fortes algoritmos de encriptação. Alguns dos dispositivos clínicos e a maioria dos *wearables* abordados anteriormente são produzidos por marcas de renome, como a *Samsung* ou a *Xiaomi*, e enviam os seus dados para *clouds* privadas e/ou públicas. Estas marcas criam e disponibilizam **APIs**, que possibilitam que os dados capturados pelos dispositivos sejam apresentados nos telemóveis, através de aplicações próprias ou aplicações externas.

2.5 Pontos Fortes e Limitações

Ao longo do documento tem-se abordado a importância e o impacto que a telemonitorização tem na saúde das pessoas e nas instituições. Sistematizando, existem vários fatores a realçar, desde logo a redução de custos. Ao recorrer à telemonitorização, o paciente irá diminuir de forma significativa o número de visitas a centros hospitalares, poupando por isso não só nos custos dos transportes, como também em possíveis despesas associados às consultas. Adicionalmente, os centros hospitalares terão muito menos casos de readmissões hospitalares e, uma vez que as consultas presenciais serão cada vez menos frequentes, é possível reduzir o número de horas passadas nos hospitais, o que contribui também para um ambiente de trabalho mais tranquilo e eficaz.

Outra vantagem é o reforço da democratização do acesso a cuidados médicos. Qualquer tipo de paciente deve ter acesso a todo o tipo de cuidados de saúde. Nesse sentido, a telemonitorização fortalece a relação entre o que os pacientes necessitam, onde necessitam e quando necessitam, e deste modo, pacientes de zonas rurais com acessos reduzidos, ou em estados de saúde bastante débeis, podem ter acesso a acompanhamento remoto e a cuidados de saúde, que se pretendem de qualidade para todos. Assim, não necessitam de se deslocar tantas vezes às instituições de saúde, o que permite reduzir, para além do custo, a exposição a doenças.

Mais recentemente, com a propagação da COVID-19, os centros hospitalares tornaram-se lugares críticos, onde os níveis de contágio eram extremamente elevados, e por isso, não era de todo aconselhável a deslocação, à exceção em casos de emergências. Como tal, o acompanhamento remoto tornou-se crucial e a telemonitorização reforçou ainda mais a sua importância.

Para além disso, uma vez que os dispositivos estarão sob o controlo dos pacientes, serão criados novos hábitos e rotinas associados à utilização destes aparelhos. Tais hábitos permitirão que os próprios

pacientes consigam entender melhor o seu estado de saúde, através da análise de padrões dos valores apresentados, podendo assim adaptarem-se, de modo a contribuir para melhores resultados e níveis dos seus sinais vitais.

Contudo, ainda há algumas limitações/obstáculos no que toca à telemonitorização. Para além de ser necessário um nível razoável de literacia digital para utilizar as aplicações, os dados registados pelos aparelhos devem ser transmitidos em tempo real ou quase-real, e para tal é necessária uma boa rede de comunicação. Em alguns casos, como por exemplo, na população mais idosa, muitas vezes localizada em sítios mais remotos, tal não acontece. Para contrariar os problemas de comunicação, alguns fabricantes estão a permitir que os seus aparelhos também consigam transmitir os dados através da rede móvel, ou então, que os dados fiquem armazenados até que o paciente volte a ter acesso à *internet* ou esteja em contacto presencial com o seu profissional de saúde, método que não é tão eficaz quanto a transmissão em tempo real.

Adicionalmente, em alguns casos, a fiabilidade dos dados recolhidos pelos pacientes pode ser afetada pelo funcionamento irregular dos aparelhos. Para minimizar este problema, devem-se procurar dispositivos certificados e provenientes de empresas de confiança, de modo a garantir que os dados recolhidos correspondem aos reais. Para além disso, quer o paciente, quer os profissionais de saúde responsáveis devem estar cientes do correto funcionamento dos aparelhos. No entanto, a aprendizagem e o treino, tanto no manuseamento dos dispositivos como das aplicações, resolvem apenas parcialmente o problema. Em termos de segurança, é necessário proceder à devida proteção de dados. Atualmente, com a digitalização da partilha de informação, surgiram novos mecanismos de aceder ilegalmente aos mais diversos dados. No caso da telemonitorização, é necessária uma preocupação adicional na maneira como os diferentes utilizadores conseguem obter a informação a partir das base de dados, sendo necessários mecanismos eficientes de encriptação e segurança, para que assim se garanta que fontes externas não conseguem aceder e manipular os dados privados dos pacientes.

Por fim, e não menos importante, as aplicações de telemonitorização devem ser de simples utilização e adaptadas, o máximo possível, às reais necessidades de cada paciente.

2.6 Casos de Aplicação

No sentido de desenvolver uma boa aplicação, é necessário primeiro analisar alguns exemplos de casos de sucesso na área de telemonitorização.

2.6.1 Telemonit SNS 24

O primeiro exemplo nesta secção é uma aplicação nacional, denominada Telemonit SNS 24 [1]. Trata-se de uma aplicação gratuita, oferecida pelo Serviço Nacional de Saúde, concebida para os utentes que necessitam de acompanhamento regular por indicação de um determinado profissional de saúde. Para além de conseguirem aceder aos dados pessoais de monitorização clínica e preencherem questionários de autoavaliação, os utentes podem também registar diferentes medições biométricas, quer manualmente, quer através da utilização de equipamentos *Bluetooth* previamente homologados para o efeito.

O objetivo da aplicação é fornecer uma ferramenta que permita a gestão contínua do estado de saúde do paciente, sendo o próprio responsável por realizar as leituras clínicas através dos aparelhos adquiridos. Até ao momento, a aplicação inclui apenas programas de telemonitorização para a insuficiência cardíaca congestiva, doença pulmonar obstrutiva crónica e recuperação após COVID-19. Relativamente às medições biométricas, o utente pode registar a saturação de oxigênio, frequência cardíaca, pressão arterial e temperatura, usando para tal dispositivos que tenham sido prescritos por um profissional de saúde e que sejam compatíveis para registo na aplicação.

Através da aplicação, os profissionais de saúde conseguem analisar os valores inseridos pelos utentes e, com base nessa recolha, verificar se necessitam de intervenção médica. No entanto, para ter acesso à aplicação, é necessário, previamente, receber indicação e autorização de um profissional. Tal indicação poderá ocorrer após uma consulta, após o utente ter obtido alta hospitalar e necessitar de continuar a ser acompanhado remotamente, ou qualquer outra circunstância que justifique acompanhamento remoto.

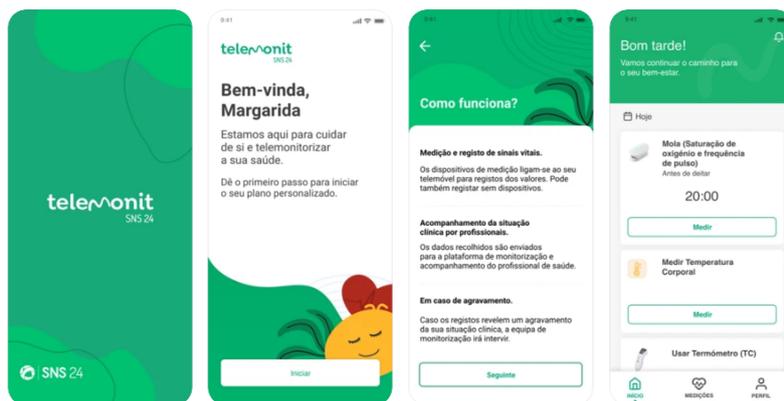


Figura 3: Telemonit SNS 24

2.6.2 HopeCare

O segundo exemplo foi desenvolvido pela empresa HopeCare [11], e é uma solução de saúde conectada para acompanhar os mais diversos tipos de pacientes, quer estes tenham problemas de saúde ligeiros ou graves.

De momento, a aplicação está adaptada para controlar não só fatores simples como humor, sono, peso e massa corporal, passos e atividade física, como também fatores mais específicos e intrínsecos a diversas doenças, como controlar a frequência cardíaca, tensão arterial, glicose, colesterol, temperatura corporal, oximetria e hemoglobina glicada. Para o controlo destas medições, é possível conectar dispositivos, plataformas ou então inserir manualmente os dados. Estes são posteriormente apresentados num *dashboard*, onde os níveis das medições são enfatizados pela utilização de cores, que auxiliam na compreensão das leituras.

De modo a ter acesso ao serviço, é necessário uma subscrição prévia e aceitação por parte da empresa, sendo que só depois destes passos serem concluídos é que as credenciais de acesso à aplicação são enviadas aos utilizadores.

A empresa HopeCare desenvolveu também o sistema HcAlert [10], que consiste num sistema de analítica aplicada à telemonitorização de dados de saúde.

Através de serviços *cloud*, a HcAlert recebe dados de saúde e alarmes em situações de emergência, de diversos fornecedores. Com base nesses dados, a aplicação cria alertas para o utilizador, para que assim seja notificado no caso de estar perante uma situação de perigo.

O principal objetivo do sistema centra-se em gerir remotamente doentes crónicos, e monitorizar os seus níveis de saúde. O sistema implementa um sistema de teleconsulta, no qual podem ser integrados profissionais de saúde e inquiridos, contribuindo assim para uma melhor triagem e decisão clínica. A aplicação possui uma interface intuitiva e de fácil utilização, com informação centralizada, segura e acessível a todos os profissionais de saúde. O produto segue um modelo de negócio *Business-to-Business*, sendo as instituições de saúde e as companhias de seguros associadas ao setor da saúde os principais clientes.

Em 2018, um projeto baseado no HcAlert esteve em funcionamento no Centro Hospitalar e Universitário de Coimbra, destinado a doentes com Doenças Pulmonares Obstrutivas Crónicas [22]. Desde a sua implementação, o projeto baixou em 50% o número de internamentos e em 30% as idas às urgências, demonstrando assim o sucesso e impacto que este tipo de soluções pode ter, quer nos pacientes, quer nos centros hospitalares.

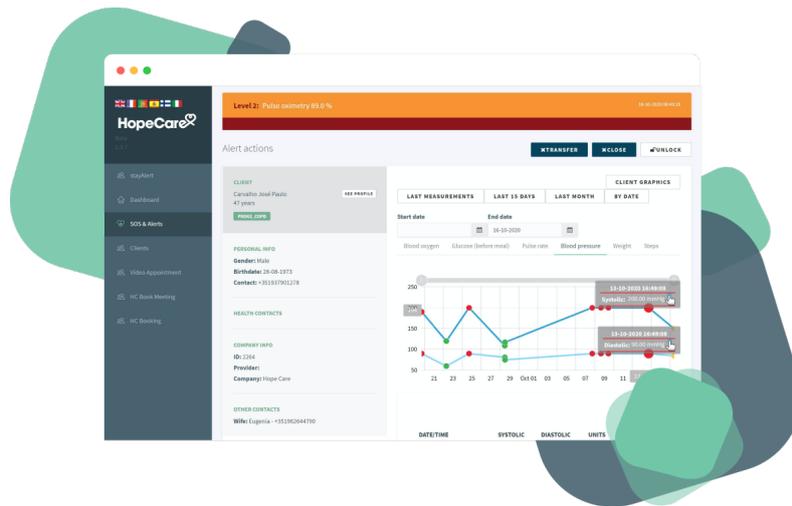


Figura 4: HopeCare

2.6.3 Homedoctor

Uma aplicação que tem vindo a ganhar destaque na área da *eHealth* é o *Homedoctor* [9]. Através da utilização do *robot* da *Homedoctor*, é possível aos pacientes monitorizarem os seus sinais vitais e armazenarem o registo de dados, para que possam ser enviados para um responsável médico, capaz de detetar possíveis problemas de saúde. Relativamente à troca de informação entre o paciente e os responsáveis de saúde, o *robot* dispõe de uma conexão automática aos hospitais, permitindo que as equipas médicas acedam aos registos de forma praticamente instantânea.

Adicionalmente, o serviço dispõe de uma plataforma que integra inteligência artificial e é baseada em engenharia semântica, designada como *BRAIN*. Esta plataforma permite que os dados de cada paciente sejam processados de modo que, em conjunto com uma série de algoritmos, se possam desenvolver recomendações personalizáveis e que se adaptem a cada paciente. Uma outra funcionalidade interessante, é que permite, em caso de emergência agendar uma consulta na hora, podendo-se se necessário passar a prescrição eletrónica para ser usada em qualquer farmácia.

homedoctor

Figura 5: Homedoctor

2.6.4 Análise Comparativa das Aplicações

Uma vez referidas e analisadas algumas aplicações, é importante identificar os aspetos mais positivos de cada uma, para que se possam definir requisitos para as novas aplicações a desenvolver no âmbito deste projeto. Todas as aplicações permitem controlar um leque diverso de medições biométricas, usando para tal dispositivos inteligentes. Estes aparelhos têm a capacidade de registar os dados automaticamente após cada medição, contribuindo assim para uma melhor usabilidade e precisão no processo de medição.

Para além disso, todos os dados recolhidos são apresentados nas aplicações, de forma a facilitar a leitura e diferenciar de imediato as situações normais das mais críticas, usando-se a cor mais esverdeada para as primeiras e a mais avermelhada para as segundas. Através do valor e da cor torna-se muito mais fácil, tanto para o doente como para o profissional de saúde saber quais as situações com que se deve preocupar e priorizar a sua resolução se for caso disso.

A maioria das aplicações permite disponibilizar questionários e definir um conjunto de tarefas diárias para os pacientes, porém, apenas algumas, como o *HomeDoctor* e a *HcAlert*, oferecem a capacidade de agendar teleconsultas, que contribuem para uma melhor triagem e decisão clínica. Com algumas nuances e à execução da prescrição eletrónica, todas estas funcionalidades estão presentes no produto de telemonitorização **B2B** da *Altice Labs*, o *SmartAL*⁶.

2.7 Solução SmartAL

Após referidos e analisados alguns exemplos de aplicações desenvolvidas por outras empresas, é agora necessário analisar o que tem vindo a ser desenvolvido pela *Altice Labs* na área da *eHealth*, mais especificamente na vertente da telemonitorização, e qual o seu impacto no desenvolvimento do presente projeto.

Tal como já referido, a *Altice Labs* desenvolveu uma solução de telemonitorização, denominada *SmartAL*. O produto *SmartAL* atualmente disponível no mercado segue um modelo de negócio *Business-to-Business (B2B)*, ou seja, é vendido a instituições e só depois estas disponibilizam o serviço aos seus clientes/doentes, geralmente de forma gratuita.

Esta solução inclui três tipos de aplicações para diferentes perfis e interfaces. A aplicação *Web* permite aos cuidadores formais visualizarem os dados monitorizados em tempo real e agendarem tarefas diárias e alertas, para posteriormente serem acedidas e executadas pelos pacientes. Esta aplicação também está disponível para outros perfis, tais como cuidador informal, administrativo e instalador. A aplicação

⁶ <https://www.alticelabs.com/products/ehealth-smart-assisted-living/>

móvel que corre em *Android* e *IOS* é ideal para os doentes, pois é mais simples, *user-friendly* e permite recolher medições de dispositivos *bluetooth* e outros *wearables*. Por outro lado, a aplicação para *Android TV*, permite a utilizadores, eventualmente menos digitalizados, efetuarem as operações mais importantes disponíveis na aplicação móvel, isto é, recolher e visualizar medições provenientes de vários tipos de dispositivos, responder a questionários e visualizar vídeos. Através de todas estas aplicações, a solução *SmartAL* almeja que os vários tipos de utilizador/doentes alcancem um elevado nível de independência, autonomia, dignidade, e segurança, dentro e fora das suas habitações.

Para além do produto **B2B**, está em desenvolvimento uma outra solução *Business-to-Consumer* (**B2C**) em linha com a nova tendência de *uberização*⁷ de serviços que pretende facilitar ainda mais o acesso a cuidados de saúde.

O intuito desta solução é permitir que serviços de saúde sejam fornecidos diretamente a cuidadores e doentes, sem estarem associados a uma instituição específica. Este protótipo segue as novas tendências tecnológicas, usando para tal, uma arquitetura de micro-serviços em ambiente de *cloud* pública. Apesar de estar em desenvolvimento, a solução já apresenta uma base sólida, estando a maioria dos serviços genéricos necessários ao funcionamento de uma aplicação *Web* definidos e desenvolvidos. Como tal, para o presente projeto, será considerada a arquitetura apresentada na Figura 6, já existente na solução **B2C**, adaptando-a para o contexto do serviço de telemonitorização a desenvolver.

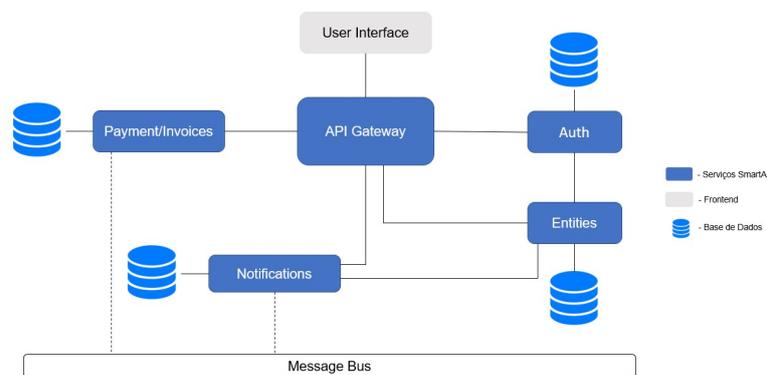


Figura 6: Arquitetura SmartAL

Dela, constam os seguintes micro-serviços:

- **API Gateway** - Oferece uma interface que torna possível a comunicação entre os clientes externos

⁷ Uberização na área da saúde refere-se à oferta de serviços de saúde diretamente através da Internet a doentes e profissionais a qualquer hora e em qualquer lugar. Profissionais de saúde passam a poder contactar e acompanhar os seus doentes sempre e quando precisam, usando para isso os meios tecnológicos necessários.

e os vários micro-serviços existentes. A *gateway* recebe os pedidos dos clientes, envia-os para os restantes micro-serviços da aplicação e, quando os micro-serviços tiverem processado os pedidos, o *gateway* devolve a informação aos respetivos clientes externos, como por exemplo a aplicação *web* de telemonitorização a desenvolver.

- **Payment/Invoices** - Micro-Serviço responsável por controlar a faturação e os pagamentos a efetuar relativos à subscrição das aplicações e ao pagamento por evento, tais como teleconsultas. Este serviço será apenas usado a título experimental, pois o foco do trabalho não está no modelo de negócio.
- **Notifications** - Micro-Serviço que permite o envio de *email*, *SMS* e notificações, e está à escuta nos diferentes canais, para quando outros micro-serviços publicarem as suas notificações no *Message Bus*. Assim, o serviço de notificações armazena a informação que foi publicada e posteriormente, sempre que necessário, irá enviar *emails*, *SMSs* e *push notifications*
- **Message Bus** - Serviço que permite a comunicação assíncrona entre alguns dos micro-serviços, através do padrão *publish-subscribe*.
- **Entities** - Serviço responsável por gerir os utilizadores e os respetivos perfis do sistema.
- **Auth** - Responsável pela autenticação dos utilizadores no sistema, utilizando o *Keycloak*⁸ como *identity provider* (IdP), que guarda e gere as credenciais dos utilizadores do sistema.

Após esta breve introdução às duas soluções SmartAL, pode então concluir-se que este trabalho terá como base, por um lado a solução *SmartAL*, como inspiração à conceção dos serviços de telemonitorização, mas a desenvolver de forma mais modular e flexível, e por outro, o protótipo **B2C** que servirá para adquirir conhecimento sobre abordagens mais inovadoras em termos de modelos de negócio, tecnologias e arquitetura em micro-serviços. Combinando o conhecimento adquirido em ambas as soluções, será possível desenvolver o projeto final.

2.8 Conclusão

Ao longo do capítulo foram abordados conceitos importantes que contribuíram para um melhor entendimento da área da saúde, mais especificamente, no ramo da *eHealth*. Foi também descrita a funcionalidade e estrutura típica de um serviço de telemonitorização, assim como a forma como as diferentes

⁸ Keycloak é um *software open source* com base num servidor *JBoss*, utilizado para gestão de processos de autenticação e autorização.

camadas da arquitetura comunicam entre si para fazer chegar a informação aos utilizadores. Associado à estrutura, referiu-se também quais as principais doenças crónicas a serem consideradas nos serviços de telemonitorização atuais, bem como os principais dispositivos utilizados.

Finalmente, foi feito um levantamento de algumas aplicações *eHealth* já existentes, quer na *Altice Labs*, quer em empresas externas, e estudadas as funcionalidades de cada uma, de modo a, no processo de desenvolvimento do projeto atual, ter em consideração tais aspetos e assim desenvolver uma aplicação o mais completa possível.

Capítulo 3

Modelação e Arquitetura

O desenvolvimento da aplicação foi definido por diferentes etapas, sendo a primeira, e não menos importante, a identificação dos requisitos do sistema. Esta identificação permite que, antes de se iniciar o desenvolvimento da solução, estejam bem definidas as regras e conceitos básicos necessários ao funcionamento correto da aplicação, para que assim se garanta que todas as funcionalidades pretendidas estejam implementadas no sistema.

3.1 Requisitos

Um requisito consiste numa funcionalidade ou restrição no desenvolvimento do sistema, e pode ser de dois tipos, funcional e não funcional. De forma a garantir que os requisitos levantados estavam de acordo com o pretendido, procedeu-se ao levantamento de requisitos em conjunto com diferentes elementos da *Altice Labs*.

Na solução final irão existir dois tipos distintos de utilizadores, sendo estes o médico e o utente. O médico representa um profissional de saúde responsável por vigiar e analisar os seus utentes, podendo depois tomar as decisões necessárias consoante a evolução ou não de um cada um destes. Por sua vez, o paciente é responsável por realizar as suas próprias medições, de modo a manter os seus responsáveis de saúde o mais a par possíveis do seu estado de saúde.

3.1.1 Requisitos Funcionais

Os requisitos funcionais definem quais as funções e informações que devem ser apresentadas no produto final. Uma vez que existem certos requisitos comuns aos dois tipos de utilizador, optou-se por agrupar os requisitos levantados por tipo de utilizador, sendo que, os requisitos relativos ao utilizador se referem quer a um paciente, quer a um médico. Foram identificados 2 requisitos relativos aos utilizadores, 12 requisitos relativos aos pacientes e 6 requisitos relativos aos médicos.

Utilizador

- **RF01** - O utilizador deve conseguir iniciar a sessão na aplicação;
- **RF02** - O paciente deve conseguir ver o seu perfil;

Paciente

- **RF03** - O paciente deve conseguir realizar medições clínicas;
- **RF04** - O paciente deve conseguir visualizar as suas medições clínicas;
- **RF05** - O paciente deve conseguir emparelhar dispositivos *bluetooth* com a aplicação;
- **RF06** - O paciente deve conseguir inserir medições manualmente;
- **RF07** - O paciente deve conseguir comentar medições realizadas;
- **RF08** - O paciente deve conseguir criar PDF com histórico de medições pretendidas;
- **RF09** - O paciente deve conseguir editar as suas *thresholds*;
- **RF10** - O paciente deve conseguir adicionar médicos para partilha de informação;
- **RF11** - O paciente deve conseguir remover médicos de partilha de informação;
- **RF12** - O paciente deve conseguir responder a questionário;
- **RF13** - O paciente deve conseguir visualizar as suas respostas a questionário;
- **RF14** - O paciente deve conseguir retomar a resolução de questionário;

Médico

- **RF15** - O médico deve conseguir visualizar os seus pacientes;
- **RF16** - O médico deve conseguir visualizar as últimas medições de um paciente;
- **RF17** - O médico deve conseguir visualizar histórico de medições de um paciente;
- **RF18** - O médico deve conseguir editar *thresholds* de um paciente;
- **RF19** - O médico deve conseguir realizar o *upload* de um questionário;

- **RF20** - O médico deve conseguir eliminar um questionário;

No Anexo [A](#) é apresentada a descrição detalhada de cada um dos requisitos acima abordados, incluindo a sua fundamentação e o nível de prioridade atribuído.

3.1.2 Requisitos Não Funcionais

Os requisitos não-funcionais, por sua vez, não descrevem quais as funcionalidades do sistema, mas antes como o sistema deve executar determinadas ações. Em seguida apresentam-se os requisitos não-funcionais recolhidos:

- **RNF01** - O sistema deve estar adaptado a várias línguas;
- **RNF02** - As páginas devem ser totalmente carregadas em menos de 1,5 segundos;
- **RNF03** - O sistema deve impedir o progresso em caso de dados inválidos;
- **RNF04** - O sistema deve ser de fácil utilização para utilizadores sem experiência;
- **RNF05** - A aplicação deve ter as funcionalidades acessíveis em poucos cliques;
- **RNF06** - O sistema deve notificar quando uma ação é concluída com sucesso;

À semelhança dos requisitos funcionais, a descrição detalhada de cada um dos requisitos não funcionais encontra-se no Anexo [B](#).

3.2 Casos de Uso

Após o levantamento de requisitos, é possível identificar aqueles que representam as principais funcionalidades do sistema, sendo por isso necessário definir os casos de uso.

Os casos de uso são utilizados para modelar e especificar os diferentes requisitos funcionais de um sistema, e cada um irá representar uma funcionalidade na solução final, sendo por isso importante especificar o mais detalhado possível cada um destes.

Na Figura [7](#) estão representados os casos de usos finais considerados no desenvolvimento do sistema. Dos 20 requisitos funcionais levantados, apenas 15 representam um caso de uso, uma vez que alguns deles podem ser associados num único caso de uso. O requisito de consultar histórico do paciente e visualizar medições de um paciente pode ser agrupado, sendo a consulta do histórico um passo adicional

na consulta das medições. Retomar a resolução de um questionário pode ser agrupado ao requisito de responder a um questionário, como sendo um passo opcional deste, caso já tenha iniciado a resolução desse mesmo questionário. A realização de uma medição clínica, a capacidade de conectar aparelhos *Bluetooth*, a possibilidade de inserir medições manualmente, e a possibilidade de comentar as medições efetuadas pode ser agrupado num único caso de uso.

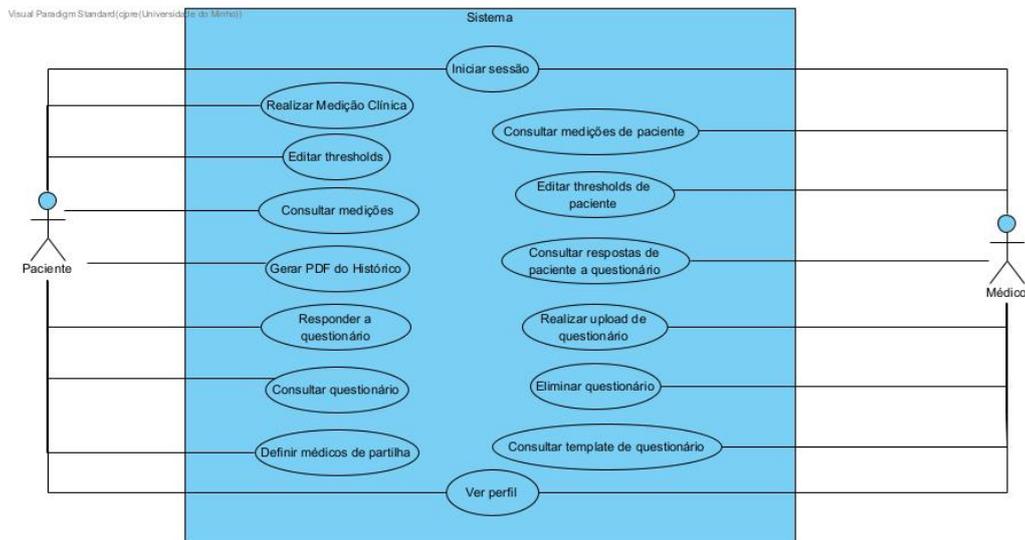


Figura 7: Diagrama de Casos de Uso

Neste capítulo, e de modo a facilitar a leitura da dissertação, apenas serão exploradas detalhadamente os dois principais casos de uso de cada tipo de utilizador, o paciente e o médico, respetivamente. Todos os restantes casos de uso poderão ser consultados no Anexo C, juntamente com as suas respetivas explicações.

3.2.1 Realizar Medição Clínica

O principal caso de uso de um paciente consiste na realização de uma medição clínica. Os pacientes do sistema possuem aparelhos *Bluetooth* capazes de realizar medições clínicas. Estes dispositivos, uma vez conectados com o sistema, poderão enviar os valores das diferentes medições efetuadas, e assim, o sistema tratará de captar estes mesmos e apresentá-los na sua interface.

Para que o sistema consiga captar os valores, terá de estar emparelhado com o dispositivo *Bluetooth*, e só depois é que o utilizador poderá iniciar a medição. Diferentes aparelhos têm diferentes instruções de utilização, e portanto é importante que o paciente esteja a par do funcionamento correto do aparelho, para que assim a medição ocorra sem problemas.

A vantagem da conexão destes dispositivos *Bluetooth* reside no facto de alguns deles conseguirem captar diferentes tipos de medições simultaneamente, isto é, é possível ter um equipamento que indique, por exemplo, não só a temperatura corporal de um paciente, como também qual a sua pulsação, sem a necessidade de recorrer a dois equipamentos distintos. Portanto, na interface do sistema, todos os valores lidos serão apresentados na interface. No entanto, o sistema também deverá permitir que, caso pretenda, o utilizador consiga inserir os valores manualmente, com a desvantagem de, em certos cenários, ter de efetuar novas medições para seleccionar todos os grupos que foram lidos pelo aparelho.

Sempre que uma nova medição é efetuada, os médicos do paciente em questão são notificados. Na Figura 8 está definida a especificação em detalhe deste caso de uso.

Use Case	Realizar medição clínica
Descrição	O paciente realiza uma medição clínica na aplicação.
Ator	Paciente
Pré-Condição	O paciente está autenticado na aplicação.
Pós-Condição	A nova medição é apresentada na secção das últimas medições.
Fluxo Normal	<ol style="list-style-type: none"> 1. O paciente seleciona a opção de adicionar nova medição. 2. A aplicação apresenta a opção de leitura <i>bluetooth</i> ou manual. 3. O paciente seleciona a opção de leitura <i>bluetooth</i>. 4. O paciente liga o aparelho <i>bluetooth</i> para que este possa ser detetado. 5. O paciente seleciona o aparelho da lista de aparelhos apresentados. 6. O sistema emparelha com o dispositivo, sinalizando o emparelhamento com um símbolo <i>Bluetooth</i> na <i>tab</i> da aplicação. 7. O sistema inicia a captação de valores do aparelho. 8. O paciente insere um comentário referente à medição efetuada. 9. O paciente submete a medição. 10. O sistema adiciona a nova medição. 11. A nova medição é apresentada na secção das últimas medições.
Fluxo Alternativo [3.1]	<ol style="list-style-type: none"> 3.1. O paciente seleciona a opção de leitura manual. 3.2. O paciente seleciona o grupo correspondente à medição que vai realizar. 3.3. O paciente introduz manualmente os valores lidos no aparelho de medição. 3.4. O paciente insere um comentário referente à medição efetuada. 3.5. Voltar a [8].

Figura 8: Use Case - Realizar medição clínica

3.2.2 Consultar medições de paciente

À semelhança do paciente, um médico com privilégios de partilha também tem acesso às medições e ao histórico de cada uma dos seus pacientes. Para aceder às medições de um paciente, o médico terá de primeiro selecionar o paciente pretendido, podendo posteriormente escolher entre ver apenas as últimas medições de cada um, ou visualizar detalhadamente todas as medições efetuadas durante um determinado período. A tabela deste caso de uso pode ser visualizada na Figura 9.

Use Case	Consultar medições de paciente
Descrição	O médico visualiza as últimas medições de um paciente.
Ator	Médico
Pré-Condição	<ol style="list-style-type: none">1. O médico está autenticado no sistema.2. O paciente já realizou medições.3. O médico encontra-se na <i>tab</i> dos seus pacientes e seleciona a opção de medições.
Pós-Condição	As últimas medições ou o histórico do paciente são apresentadas.
Fluxo Normal	<ol style="list-style-type: none">1. O médico seleciona o paciente pretendido.2. O sistema adquire a informação do paciente.3. O médico seleciona a opção de medições.4. O sistema adquire as medições desse paciente.5. As últimas medições do paciente são apresentadas.6. O médico visualiza medições.
Fluxo Alternativo 1 [6.1]	<ol style="list-style-type: none">6.1. O médico seleciona a medição que pretende visualizar.6.2. O médico seleciona a opção de gráfico6.3. O médico seleciona o período pretendido.6.4. O histórico é apresentado no ecrã.
Fluxo Alternativo 2 [6.2.1]	<ol style="list-style-type: none">6.2.1. O médico seleciona a opção de tabela.6.2.2. Voltar a [6.3]

Figura 9: Use Case - Consultar medições de paciente

3.3 Arquitetura Funcional

Uma vez definidos os casos de uso, é possível definir quais os serviços da aplicação final. Analisando cada funcionalidade, verifica-se que existem 5 componentes principais facilmente distinguíveis, e que representam serviços na arquitetura final, sendo estes *Entities*, *Notifications*, *Measurements*, *Questionnaires* e *Shared Info*.

O serviço *Entities* engloba as entidades paciente e médico, que são armazenadas de modo a possibilitar a autenticação dos utilizadores no sistema e já se encontra implementado no *SmartAI* e portanto não precisa de ser desenvolvido.

Sempre que alguma ação que mereça destaque seja executada por um paciente, é importante notificar os seus médicos. Portanto, o serviço *Notifications* será responsável por armazenar, de forma organizada, as ações realizadas. À semelhança do serviço *Entities*, também já se encontra implementado.

As medições e o histórico podem ser vistos como um único serviço, capaz de armazenar todas as medições efetuadas por um paciente, o que permitirá a criação de um histórico de medições. O serviço composto por estes dois componentes denomina-se de *Measurements*.

De modo a armazenar a relação de partilha de dados entre um paciente e um médico é necessário criar um serviço único, denominado *Shared Info*. Este serviço apenas será composto pelos identificadores de cada utilizador e irá possuir as relações entre eles.

Por fim, os questionários possuem o seu próprio serviço, o serviço *Questionnaires*, que para além de guardar os questionários, também armazena as respostas de cada paciente, por cada submissão realizada.

Na Figura 10 encontra-se representada a arquitetura final desenvolvida. A azul estão representados os serviços já implementados no *SmartAI*, e que por isso serão reutilizados, e a verde estão representados os novos serviços a serem desenvolvidos.

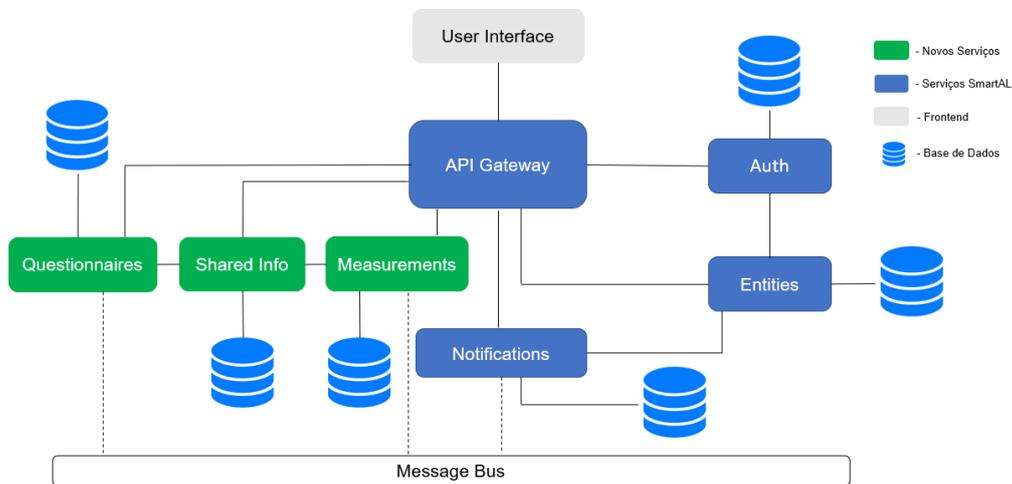


Figura 10: Arquitetura Funcional Final

3.4 Diagramas de Sequência

De forma a facilitar a compreensão da lógica associada a cada um dos *use cases* especificados, recorre-se a diagramas de sequência. Um diagrama de sequência representa as interações entre objetos, através

das mensagens que são trocadas entre eles, especificando adicionalmente qual a respetiva sequência temporal correta. Estes diagramas são compostos essencialmente por três componentes, os atores, que representam as entidades que irão interagir com o sistema final, as linhas de vida, que representam os processos existentes na execução de uma determinada ação, e as linhas horizontais, que representam as mensagens trocadas entre as linhas de vida existentes.

Tal como na secção dos casos de uso, nesta secção apenas se irá explicar devidamente o principal diagrama de sequência, quer do paciente, quer do médico, podendo os restantes diagramas e as respetivas explicações ser consultadas no Anexo D.

3.4.1 Realizar medição clínica

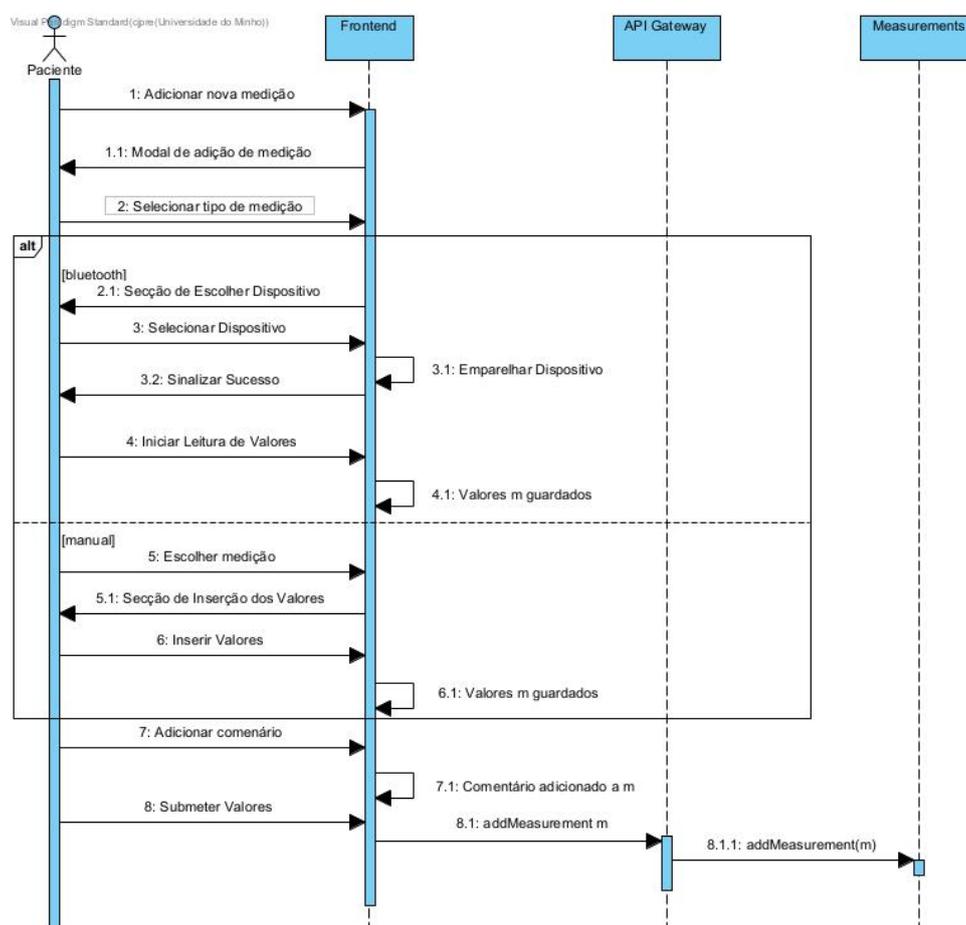


Figura 11: Diagrama Sequência - Realizar medição clínica

Como se pode observar na Figura 11, o serviço *API Gateway* recebe os pedidos provenientes da *Frontend*, e reencaminha para o respetivo serviço responsável por tratar o tipo de informação enviada. Neste caso, como se está a efetuar uma medição clínica, o pedido será reencaminhado para o serviço *Measurements*.

Também é importante referir que, como as medições clínicas podem ser efetuadas manualmente ou via *Bluetooth*, o comportamento da aplicação será diferente. No caso de uma medição *Bluetooth*, o paciente terá de emparelhar o seu dispositivo com a aplicação, e só depois os valores começam a ser lidos pelo sistema e apresentados na interface. No caso da leitura manual, o paciente terá de indicar que tipo de medição irá efetuar, e inserir os valores pretendidos. Adicionalmente, poderá ser adicionado um comentário, que possa ser útil para a interpretação dos valores. Depois de todos os dados recolhidos, a aplicação indicará à *API Gateway* o pedido de adicionar uma nova medição, que redireciona para o serviço *Measurements* e este trata de adicionar a nova medição no sistema.

3.4.2 Consultar medições de paciente

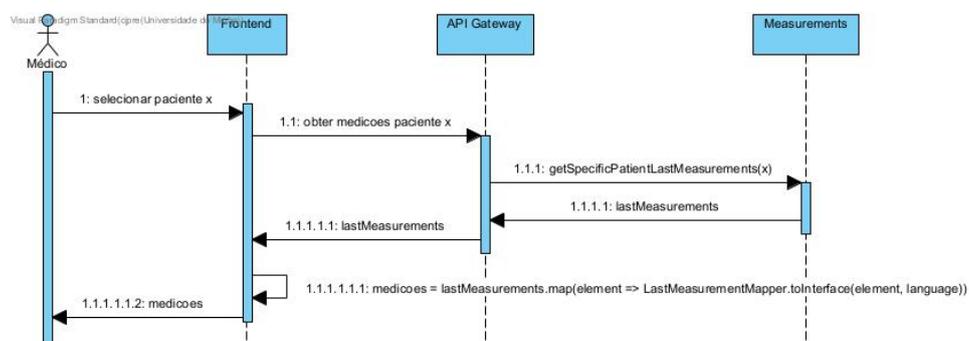


Figura 12: Diagrama Sequência - Consultar medições de paciente

Todas as medições efetuadas por cada um dos pacientes associados a um determinado médico poderão ser consultadas por este último. De modo a obter as tais medições, apenas será necessário conhecer o identificador do paciente que se pretende, e o sistema irá devolver todas as medições efetuadas por esse mesmo paciente. O diagrama de sequência pode ser visualizado na Figura 12.

3.5 Modelos de Dados

No âmbito da modelação do sistema, e de forma a servir a lógica descrita no capítulo anterior, será necessário desenvolver o modelo de dados. Uma vez que a solução é composta por vários micro-serviços, será necessário definir uma base de dados independente para cada um deles.

Os serviços previamente implementados pela *Altice Labs* recorrem a bases de dados relacionais, pelo que a mesma estratégia será adotada para os novos serviços.

Antes de desenvolver o modelo de dados para cada serviço, é importante definir quais as tabelas que serão criadas. Para cada tabela será necessário distinguir qual a sua chave-primária, que permitirá identificar inequivocamente cada registo, quais os seus atributos, que simbolizam a informação concreta de cada tabela, tais como características e os seus tipos, e, caso necessário, a referência a atributos de outras tabelas, de modo a referenciar a dependência entre elas.

Com tal em mente, desenvolveu-se um modelo Entidade-Relacionamento para cada um dos serviços especificados no capítulo anterior, onde estão especificados todas as tabelas, atributos e relacionamentos existentes.

3.5.1 Entities

O serviço *Entities* representado na Figura 13 já se encontrava previamente implementado na *Altice Labs*, e é responsável por gerir e armazenar todas as entidades da solução de telemonitorização *SmartAL*. Visto que o sistema a desenvolver irá ser disponibilizado às mesmas entidades de tal solução, ficou decidido que se iria reutilizar este micro-serviço *Entities*, não sendo por isso necessário redefinir mais nenhuma lógica associado a este.

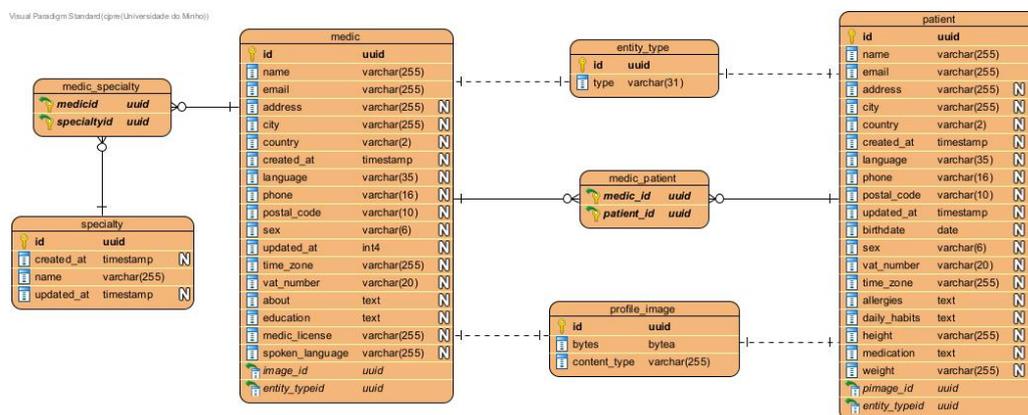


Figura 13: Modelo de Dados do Serviço *Entities*

De seguida, será efetuada uma breve explicação e contextualização de cada uma das tabelas do serviço *Entities*:

- **entity_type**: Tabela definida por um tipo e por um identificador (*id*), que serve de referência para o identificador de um utilizador. Utilizada para definir o tipo de utilizador.
- **medic**: Armazena os dados dos médicos. Cada médico possui uma lista de pacientes, daí ter uma relação com a tabela *patient*. Adicionalmente, possui uma relação *1-para-1* com a tabela *profile_image*.

profile_image, visto que um médico poderá ter apenas uma única imagem de perfil. Por último, um médico pode ter várias especialidades, daí a existência de uma relação *n-para-n* com a tabela specialty.

- **patient**: Armazena os dados dos pacientes. Um paciente pode ter vários médicos associados a si, por isso possui uma relação com a tabela medic. Também possui relação *1-para-1* com a tabela profile_image.
- **profile_image**: Armazena os dados das imagens de perfil das entidades, sendo que cada entidade poderá possuir apenas uma imagem de perfil.
- **specialty**: Armazena todas as especialidades que os médicos podem ter. É caracterizada por um identificador único e pelo nome da especialidade.
- **medic_specialty**: Armazena os identificadores dos médicos e das suas especialidades. Resulta da existência de uma relação *n-para-n* entre as tabelas medic e specialty.
- **medic_patient**: Armazena os identificadores dos médicos e dos seus pacientes. Tabela originada pela existência de uma relação *n-para-n* entre as tabelas medic e patient.

3.5.2 Notifications

À semelhança do serviço *Entities*, o serviço *Notifications* já se encontra totalmente desenvolvido na solução *SmartAI*, pelo que será reaproveitado para o presente projeto. Este serviço permite enviar notificações aos utilizadores, sempre que alguma ação que mereça destaque seja efetuada. Quando um utilizador se autentica, é gerado um *FirebaseToken*, com a informação de autenticação desse utilizador, que poderá ser utilizado para aceder às suas notificações. Na Figura 14 encontra-se representado o modelo de dados do serviço *Notifications*.

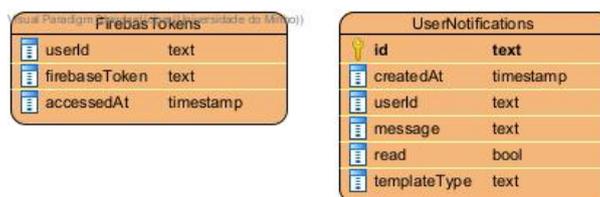


Figura 14: Modelo de Dados do Serviço Notifications

De seguida, será efetuada uma breve explicação e contextualização de cada uma das tabelas do serviço *Notifications*:

- **FirestoreTokens:** Armazena os *tokens* dos utilizadores, bem como um *timestamp* em que o *token* foi gerado.
- **UserNotification:** Armazena todas as notificações do sistema. Para cada notificação, existe um atributo *templateType* que define como a presente *message* irá ser disponibilizada ao utilizador associado ao atributo *userId*. No contexto da aplicação, este tipo pode ser referente a medições efetuadas e a resoluções de questionários submetidas. Através do atributo *read*, é possível saber se o utilizador já acedeu à mesma notificação.

3.5.3 Measurements

Na Figura 15 encontra-se representado o modelo de dados do serviço *Measurements*. Um paciente, ao utilizar os equipamentos *Bluetooth* e iniciar a captura de valores, estará a realizar uma medição, que é constituída por uma ou mais medidas. Cada uma destas medidas tem um tipo de dados associado a si, sendo cada tipo caracterizado por um código, uma unidade e uma *flag* denominada *numdecimal* que simboliza a presença ou não de números decimais no valor a ser lido. Por sua vez, cada tipo de dados pertence a um determinado grupo, que pode ter vários tipos de dados. Adicionalmente, e de modo a perceber se os valores das medições realizadas se encontram dentro ou fora do normal, é necessário definir os limites positivos e negativos de cada medição, podendo estes ser pessoais ou globais. Estes limites denominam-se *thresholds* e, caso um paciente pretenda, pode ter os limites de uma medição ajustados exclusivamente a si.

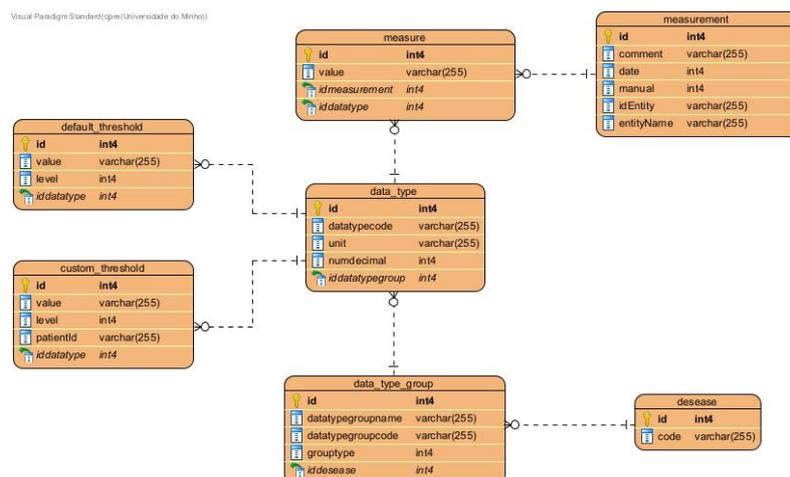


Figura 15: Modelo de Dados do Serviço Measurements

De seguida, será efetuada uma breve explicação e contextualização de cada uma das tabelas do

serviço *Measurements*:

- **measurement**: Armazena todas as medições efetuadas pelos pacientes. Possui uma relação de 1 para n com a tabela *measure*, visto que uma medição pode ser composta por várias medidas. Adicionalmente, o atributo *idEntity* permite identificar qual o paciente que realizou uma determinada medição.
- **measure**: Armazena cada medida efetuada. Possui um atributo de referência para a tabela *measurements* a que pertence, e um atributo para a tabela *data_type* referente ao tipo da medida em questão.
- **data_type**: Armazena todos os tipos de dados do sistema. Uma medida possui um único tipo de dados, e por sua vez, este pertence a apenas um grupo de tipo de dados. Para cada tipo de dados, existe um conjunto de limites *default* e limites customizáveis, daí a existência de relações 1 para n com as tabelas *default_threshold* e *custom_threshold*,
- **data_type_group**: Armazena todos os grupos de tipos de dados. Possui uma relação 1 para n com a tabela *data_type*, visto que cada grupo possui vários tipos de dados.
- **default_threshold**: Armazena os limites *default*. Cada limite é composto por um valor e um nível. Quando o valor de uma medida é comparado com os limites de um tipo de dados, é possível entender se a medida se encontra dentro ou fora da norma.
- **custom_threshold**: Armazena os limites customizáveis. Idêntica à tabela *default_threshold*, mas com a adição do atributo *patientId*, que faz referência ao identificador do paciente ao qual o limite customizável pertence.
- **disease**: Armazena todas as doenças existentes. Uma doença pode estar associada a vários *data_type_groups*.

3.5.4 Shared Info

Todas as ações realizadas por um paciente devem ser disponibilizadas e partilhadas com os médicos da sua escolha. Para esse efeito, foi desenvolvido o serviço *Shared Info*, que é essencialmente composto pelos identificadores de cada uma das entidades, como se pode verificar na Figura 16.

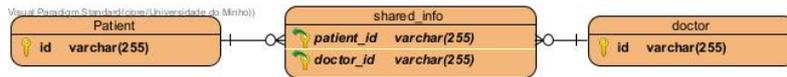


Figura 16: Modelo de Dados do Serviço Shared Info

De seguida, será efetuada uma breve explicação e contextualização de cada uma das tabelas do serviço *Shared Info*:

- **patient**: Armazena os identificadores de todos os pacientes no sistema. Um paciente pode estar associado a vários médicos.
- **doctor**: Armazena os identificadores de todos os médicos no sistema. Um médico pode estar associado a vários pacientes.
- **shared_info**: Tabela formada pelo estabelecimento da relação *n* para *n* entre as tabelas patient e medic.

3.5.5 Questionnaires

De modo a obter mais informações acerca do estado de saúde dos seus pacientes, os médicos disponibilizam questionários. Um questionário é composto por um conjunto de grupos, questões e respostas, e pode ser visualizado e respondido múltiplas vezes, por cada paciente. Estas duas hipóteses fazem com que não seja apenas necessário armazenar o questionário em si, como também as respostas escolhidas por cada paciente, em cada uma das vezes que respondeu ao questionário.

Adicionalmente, ficou estabelecido que a organização de um questionário pode ter dois tipos. O primeiro tipo, e mais habitual, consiste na ordenação das perguntas pela posição onde estas devem estar. Já o segundo tipo não é tão linear, sendo que a próxima questão a ser apresentada a um utilizador irá depender da resposta selecionada na questão anterior. O modelo de dados final pode ser visualizado na Figura 17.

De seguida, será efetuada uma breve explicação e contextualização de cada uma das tabelas do serviço *Questionnaires*:

- **questionnaire**: Armazena os dados relativos aos questionários. É caracterizado por um nome, e possui uma lista que grupos, daí a existência da relação 1 para *n* com a tabela group. Um questionário pode estar associado a várias sessões.

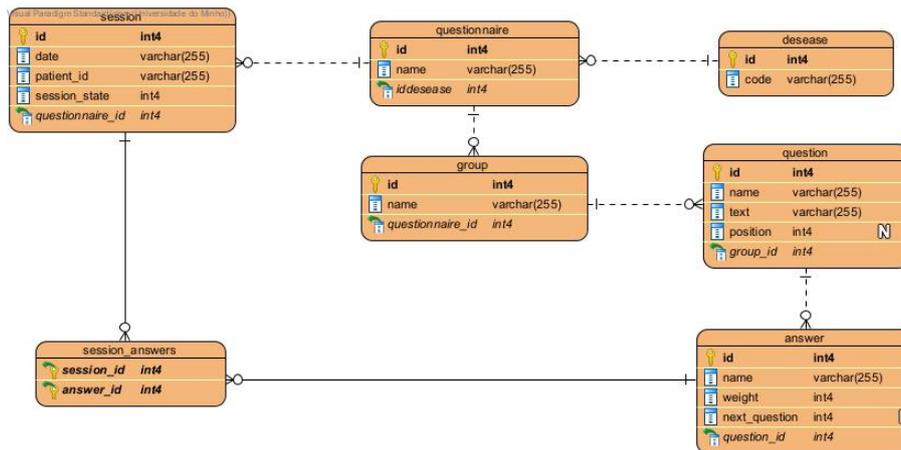


Figura 17: Modelo de Dados do Serviço Questionnaires

- **session:** Armazena os dados relativos às sessões criadas. Uma sessão corresponde à realização de um questionário por parte de um paciente, e está sempre associada a um questionário. O atributo *patient_id* faz referência ao identificador do paciente que criou a sessão. Cada sessão pode ter várias respostas associadas a si.
- **group:** Armazena os dados relativos aos grupos dos questionários. Um grupo está associado a um único questionário e cada grupo possui várias questões, daí a existência da relação 1 para *n* com a tabela question.
- **question:** Armazena a informação de todas as questões. Cada questão é composta por várias respostas e pode ou não ter um atributo referente à posição da questão no questionário.
- **answer:** Armazena todas as respostas possíveis. É caracterizada por um nome, um peso e pode ou não possuir um identificador para o índice da próxima questão. Cada resposta pode ter várias sessões associadas a si.
- **session_answers:** Armazena os identificadores das sessões e respostas existentes. Resulta da existência da relação *n* para *n* entre as tabelas session e answer.
- **disease:** Armazena todas as doenças existentes. Uma doença pode estar associada a vários questionários.

3.6 Conclusão

O presente capítulo centrou-se na modelação e arquitetura final do sistema a desenvolver. Abordaram-se os requisitos funcionais e não-funcionais levantados, os casos de uso que transitaram para o sistema final, a arquitetura funcional do sistema, o modelo de dados de cada um dos serviços que irão estar presentes no sistema, e por fim, os diagramas de sequência associados aos casos de uso definidos, que permitem assim melhor entender o fluxo de cada um.

Sendo esta a etapa que antecede o início do desenvolvimento da aplicação, foi bastante importante para o desenvolvimento eficaz do sistema final.

Capítulo 4

Implementação do Sistema

Após completado o processo de especificação e modelação do sistema, foi possível prosseguir com a implementação do sistema. Ao longo deste capítulo, serão abordadas as tecnologias e ferramentas utilizadas e o porquê de terem sido selecionadas, como foram implementados os micro-serviços especificados no capítulo anterior e as interfaces desenvolvidas. Por último, será abordada a maneira como se procedeu à validação e *deployment* da aplicação.

4.1 Ferramentas e Tecnologias

A aplicação está dividida, essencialmente, em três componentes distintas, *frontend*, *backend* e base de dados. Para cada uma destas componentes, existe um vasto leque de opções que podem ser viáveis para utilização no sistema. Porém, a *Altice Labs* executou um trabalho comparativo de análise de tecnologias e ferramentas aquando da definição da arquitetura de micro-serviços base. Tendo em conta este trabalho, reutilizou-se ao máximo os recursos tecnológicos existentes. Adicionalmente, há um conjunto de ferramentas e tecnologias que facilitam a gestão de projetos desta escala.

Aquando de tal escolha, foram colocados alguns requisitos à implementação do sistema. De modo a facilitar o processo de desenvolvimento, decidiu-se optar pela utilização de tecnologias usadas na *Altice Labs*, tais como *React*, que é o mais utilizado para desenvolvimento da *frontend*, e *PostgreSQL* para a base de dados. Um outro requisito foi utilizar sempre que possível ferramentas e tecnologias com extensa documentação, e com um uso significativo no mundo profissional. A *Spring Framework* é uma dessas tecnologias, utilizada para desenvolvimento de **APIs**. Uma vez que seriam utilizados dispositivos *Bluetooth* era necessário recorrer a uma **API** externa que possibilita-se a conexão a tais aparelhos, tendo para tal se escolhido a *Web Bluetooth API*. Ferramentas e tecnologias como *Git*, *Docker* e *Google Cloud* foram escolhidas para processos de gestão de código e de *deployment*.

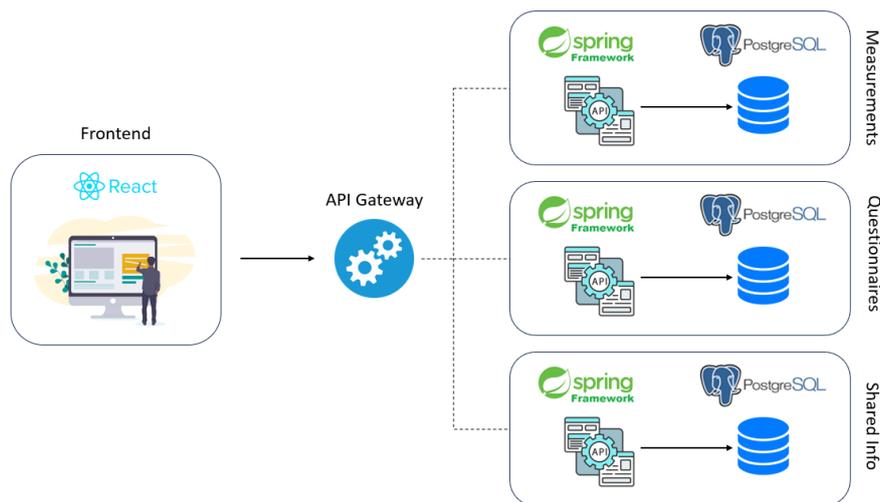


Figura 18: Ferramentas e Tecnologias

Na Figura 18 está representada a principal composição tecnológica do projeto, onde se tem a componente *frontend* desenvolvida em React, cada um dos serviços desenvolvidos utilizando a Spring Framework, e a respetiva base de dados associada a cada um destes em PostgreSQL.

De seguida serão abordadas e explicadas as principais ferramentas e tecnologias referidas anteriormente, que foram bastante importantes para o desenvolvimento do sistema.

React¹ é uma biblioteca de *JavaScript* utilizada para a programação e desenvolvimento de interfaces. Estas interfaces são formadas por diversos componentes, que possuem o seu próprio estado e que, quando associados com outros componentes, permitem criar interfaces mais complexas, geralmente denominadas de páginas. O facto da lógica de cada componente ser definida nele próprio, permite que as aplicações *React* sejam facilmente escaláveis, um dos requisitos pretendidos no atual sistema.

A nível da linguagem, optou-se pela utilização de TypeScript², uma linguagem baseada em *JavaScript*³, mas que adiciona *static typing*, que consiste na atribuição e verificação de tipos às variáveis e objetos em tempo de compilação, contribuindo não só para um melhor controlo de código, como também para uma melhor documentação deste.

A Spring framework⁴ assenta principalmente em duas características, sendo estas a Dependency Injection e o Aspect Oriented Programming. A primeira consiste no processo através do qual os objetos definem as suas dependências, ou seja, os objetos com os quais interagem, através de argumentos passados no construtor de cada objeto. Quando um *bean* é criado, essas dependências são injetadas. A segunda

¹ <https://react.dev/>

² <https://www.typescriptlang.org/>

³ <https://www.javascript.com/>

⁴ <https://spring.io/>

complementa a programação orientada a objetos, permitindo a modularização de gestão de transações.

O *PostgreSQL*⁵ é um sistema de gestão de base de dados que requer baixa manutenção, apresenta um desempenho bastante elevado e é robusto, tornando-o assim uma das bases de dados relacionais mais utilizadas atualmente. A preferência pela sua utilização advém não só do facto de ser *open-source*, e apoiada por mais de 20 anos de desenvolvimento, como também pelo facto de ser utilizada nos serviços do *SmartAL*, o que facilita o futuro *deployment* da aplicação e dos seus serviços.

Uma vez que uma das funcionalidades pretendidas residia no emparelhamento de dispositivos *Bluetooth* com o sistema, e dado o facto da aplicação ser utilizada em *browsers*, foi necessário pesquisar bibliotecas ou **APIs** que oferecessem tal funcionalidade.

Após alguma pesquisa, optou-se por utilizar a *Web Bluetooth API*⁶. Apesar de servir o propósito pretendido, esta **API** apresenta alguns problemas, sendo de destacar a escassa documentação disponível *online*, e a falta de compatibilidade com alguns *browsers*, tais como *Firefox* e *Safari*. Não obstante, e como cumpria os requisitos pretendidos, decidiu-se pela sua utilização.

O *Docker*⁷ é uma plataforma de *software* cujas funcionalidades passam por criar, executar e partilhar aplicações, usando para tal *containers*. Este utiliza o *kernel* do *Linux* para separar os processos, de modo a estes poderem ser executados de forma independente. Esta independência cria a noção de *containers*, onde é possível ter vários processos e aplicações a correr independentemente, simultaneamente.

Os *containers* fornecem um modelo de implementação baseado em imagens. De modo a construir a imagem de um *container*, é necessário desenvolver o ficheiro *DockerFile* que contém todas as instruções para a criação de um *container*. Quando as imagens são executadas, dão então origem a um *container*.

Para além das vantagens associadas a esta ferramenta, a própria *Altice Labs* utiliza *Docker* para o *deployment* das suas aplicações, e portanto optou-se por usar esta tecnologia.

De modo a testar os serviços desenvolvidos, recorreu-se ao *Postman*⁸. O *Postman* é uma plataforma que permite definir e testar **APIs**, de forma simplificada. Uma das vantagens desta ferramenta é a possibilidade de catalogar cada um dos pedidos *http* realizados, e armazená-los, para que assim se seja visualmente mais fácil distinguir todos os pedidos efetuados.

O *RabbitMQ*⁹ é um gestor de filas, onde são definidas filas de espera, às quais as aplicações se ligam para enviar e receber mensagens. Esta tecnologia implementa o protocolo *Advanced Message Queuing*

⁵ <https://www.postgresql.org/>

⁶ https://developer.mozilla.org/enUS/docs/Web/API/Web_Bluetooth_API

⁷ <https://aws.amazon.com/pt/docker/>

⁸ <https://www.postman.com/>

⁹ <https://www.rabbitmq.com/>

*Protocol*¹⁰, um protocolo da camada aplicacional que é definido pela orientação das mensagens, *queing*, *routing* e segurança.

A *Google Cloud*¹¹ é uma plataforma que oferece apoio ao *deployment* de serviços, permitindo gerir dados e aplicações de forma eficaz, segura e relativamente simples. Uma vez que a *Altice Labs* recorre a esta plataforma para fazer *deployment* das suas aplicações, optou-se por seguir a mesma estratégia para o presente projeto.

O *Git*¹² é uma ferramenta de controlo de *source-code*, utilizada para detetar mudanças que sejam feitas neste, e permitir que vários utilizadores trabalhem em conjunto, de forma não-linear. Tal advém do facto de ser possível criar repositórios, que contém os ficheiros de um determinado projeto, e onde vários utilizadores podem criar diferentes versões e trabalhar simultaneamente, cada um na sua versão, e quando for necessário, fazer *merge* da sua versão com as restantes, para que assim os outros utilizadores tenham acesso ao código desenvolvido.

A *Altice Labs* utiliza o *Git* para a gestão dos seus projetos, e portanto, esta ferramenta também foi utilizada para o desenvolvimento do presente projeto.

4.2 Implementação

A parte mais importante e longa do desenvolvimento do sistema consistiu na sua implementação. Tal como se tem referido ao longo do documento, o sistema final é constituído por uma *frontend* e três serviços (**APIs**) principais. Ao longo desta secção irão ser abordadas e explicadas as estruturas de cada um.

4.2.1 Serviços

Cada serviço é responsável por tratar da lógica, processamento de dados e interações com as diferentes bases de dados. No sistema desenvolvido, os serviços utilizados foram os enunciados previamente, porém, nesta secção, apenas se irá dar ênfase aos serviços desenvolvidos no contexto da aplicação atual, sendo estes o *measurements*, *questionnaires* e *shared-info*.

Apesar de cada serviço ser independente dos restantes, a sua estrutura é bastante semelhante. Tal como foi abordado na secção das tecnologias utilizadas, cada um dos serviços foi desenvolvido utilizando a *Spring Framework*. De modo a tirar o maior proveito desta *framework*, dividiu-se cada projeto em

¹⁰ <https://www.amqp.org/about/what>

¹¹ <https://cloud.google.com/>

¹² <https://git-scm.com/>

diferentes camadas, sendo estas a camada controller, a camada service e a camada repository.

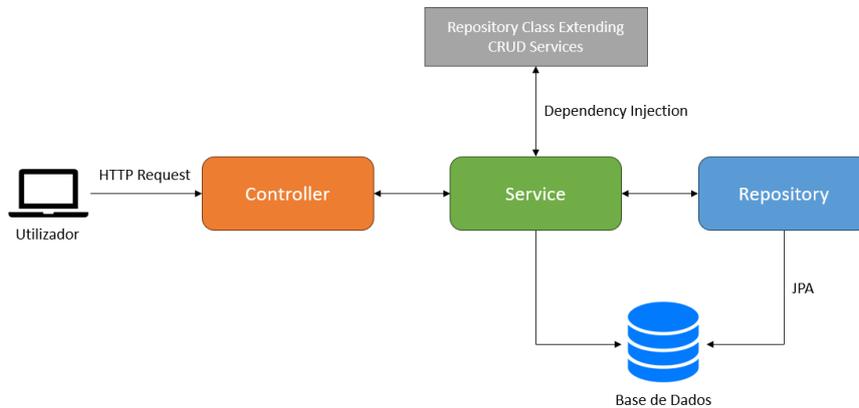


Figura 19: Fluxo de um Pedido

A Figura 19 contém o fluxo normal de um pedido efetuado a cada serviço. Um cliente, ao utilizar a aplicação, efetua um pedido *HTTP*, que é recebido pela camada controller. Esta camada é responsável por fazer o mapeamento do pedido, isto é, descobrir que tipo de pedido se trata e como devem ser tratados os dados enviados nesse mesmo pedido. Após o mapeamento efetuado, a camada irá adquirir a lógica de negócio necessária à camada service, responsável por armazenar toda a lógica de negócio necessária. Caso seja necessário o processamento/armazenamento de dados, a camada service irá aceder à camada repository, responsável por fornecer mecanismos de armazenamento, inserção e remoção em objetos. De modo a proceder a este armazenamento, é utilizada a especificação *Spring Boot JPA*, que gere os dados relacionais numa aplicação *Java* e permite aceder e persistir dados entre objetos, declarados como entities, e as base de dados relacionais.

Measurements

O serviço *Measurements* é composto por seis *controllers*, cada um associado ao tratamento específico de uma entidade. Na Figura 20 está representada a implementação do serviço *measurements*. Os componentes principais deste serviço são as classes cuja nomenclatura termina em *controller*, *service* e *repository*. Adicionalmente, definiram-se pastas que não foram ainda abordadas.

A pasta *Config* contém os ficheiros de configurações a executar quando o sistema é iniciado, sendo estes relativos ao envio das notificações e à definição dos endereços que podem aceder ao serviço. A pasta *Schemas* possui classes configuradas para receber os dados provenientes da *frontend* e assim conseguir aceder aos atributos dos dados de forma mais simples. Adicionalmente, existe a necessidade

de configurar um ficheiro *application.properties*, que possui a definição de todas as variáveis necessárias para se estabelecer conexão com as bases de dados configuradas.

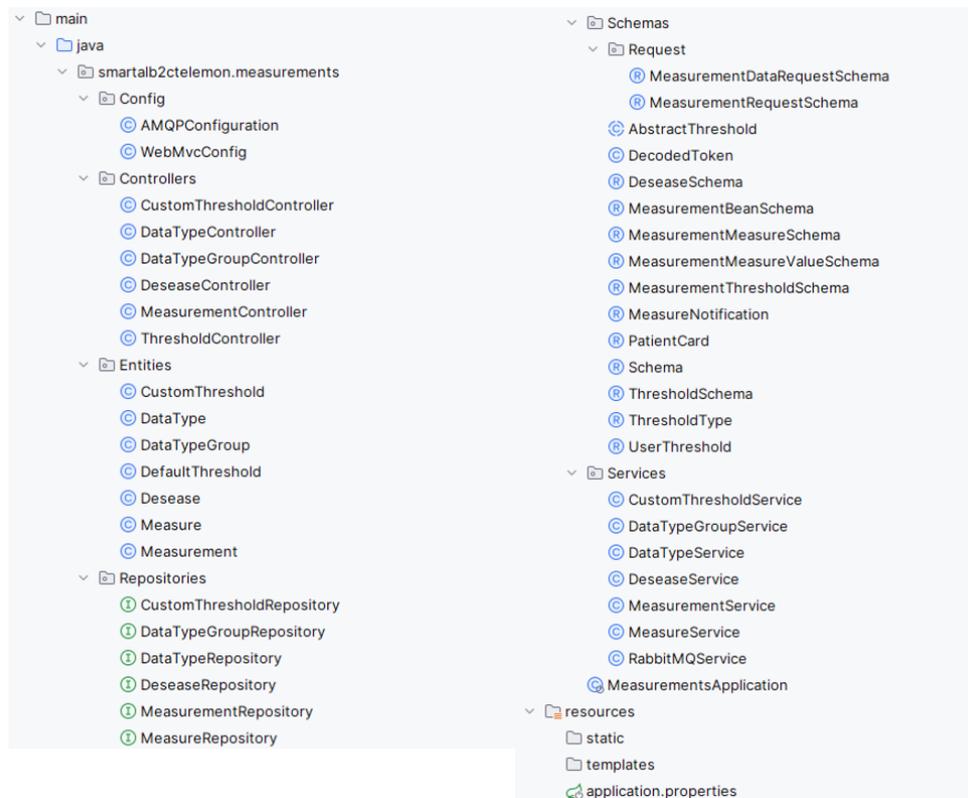


Figura 20: Implementação do Serviço Measurements

De modo a perceber como cada componente neste serviço se relaciona, será analisado em especial o exemplo da Figura 21, onde está representada o diagrama de classes do *MeasurementController*. A cada um dos métodos definidos no *controller* está associado um *request mapping*, que permite definir qual o endereço a utilizar caso se pretenda aceder a esse mesmo *controller*. No presente exemplo, o *controller* irá aceder a vários *services*, que possuem a lógica de negócio associada aos diferentes métodos. Cada *service* possui um *repository*, que serve para realizar operações na base de dados. Adicionalmente, destaca-se o *RabbitMQService*, responsável pelo processo de tratamento de envio de notificações, sempre que necessário.

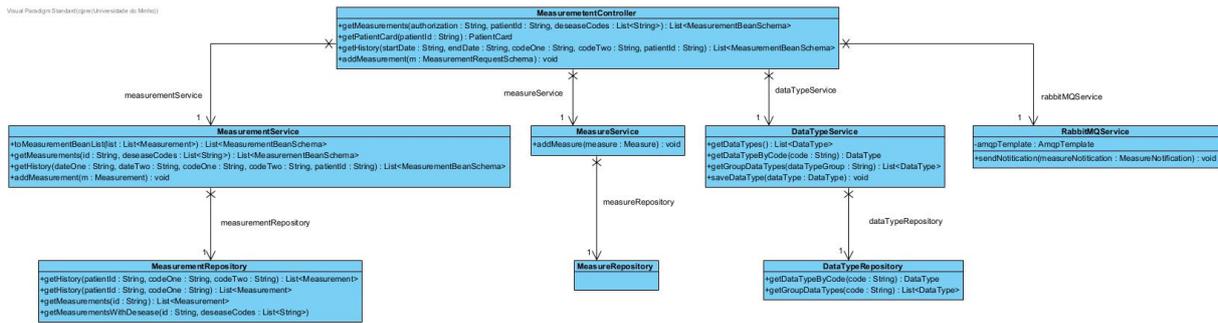


Figura 21: Estrutura MeasurementController

Um dos *endpoints* definidos pode ser observado na Figura 22, tendo como objetivo efetuar a adição de uma medição ao sistema. O primeiro passo passa por descobrir quais os médicos associados ao paciente que está a efetuar a medição. Para tal, será necessário efetuar um pedido ao serviço *Shared Info*, tal como se pode observar na linha 125 da Figura. Uma vez obtidos os médicos, é necessário iterar por todos os valores presentes no objeto recebido e criar uma medição com cada um desses valores, para posteriormente se adicionarem ao sistema. Por fim, é enviada a notificação a todos os médicos desse mesmo paciente, a indicar que uma nova medição foi efetuada.

```

117
118 @PostMapping(path = @("/measurement"))
119 @Controller
120 public void addMeasurement(@RequestBody MeasurementRequestSchema m) {
121     Measurement measurement = new Measurement();
122     List<Measure> measures = new ArrayList<>();
123
124     RestTemplate restTemplate = new RestTemplate();
125     Map<String, Object> params = new HashMap<>();
126     params.put("patientId", m.getIdEntity());
127     List<String> result = restTemplate.getForObject("url:sharedInfoAPI + "?patientId={patientId}", List.class, params);
128
129     m.values().forEach(value -> {
130         String dataTypeCode = value.getDataType();
131         DataType dataType = this.dataTypeService.getDataTypeByCode(dataTypeCode);
132         Measure measure = new Measure(value.value(), dataType, measurement);
133         measures.add(measure);
134         this.measureService.addMeasure(measure);
135     });
136
137     this.rabbitMQService.sendNotification(new MeasureNotification(m.getIdEntity(), result, measures.get(0).getDataType().getDatatypegroup().getDatatypegroupcode()));
138
139     measurement.setIdEntity(m.getIdEntity());
140     measurement.setEntityName(m.getFormalUsername());
141     measurement.setMeasures(measures);
142     measurement.setComment(m.values().get(0).comment());
143     measurement.setDate(m.date());
144     measurement.setManual(m.manual());
145     this.measurementService.addMeasurement(measurement);
146 }

```

Figura 22: Endpoint para adicionar Medição

Para além do exemplo acima, foram definidas mais rotas, cada uma delas necessária para se cumprir os requisitos funcionais definidos.

De seguida, serão apresentadas todas as rotas definidas, bem como quais os requisitos funcionais associadas a cada uma:

- **GET /measurement/measurements** - Pedido que retorna as Measurements de um paciente, podendo estas estar filtradas por doenças. Associada ao requisito funcional **RF04**.
- **GET /measurement/patientcard** - Pedido que devolve as medições de um paciente, e quantas estão a vermelho e a amarelo. Associado ao requisito funcional **RF15**.
- **GET /measurement/history** - Pedido efetuado para obter o histórico de um paciente, para um dado período e para determinados grupos. Associado ao requisito funcional **RF17**.
- **GET /datatype/datatypes** - Pedido efetuado para obter todos os datatypes presentes no sistema. Associado ao requisito funcional **RF17**.
- **GET /datatype/groupdatatypes** - Pedido efetuado para obter todos os datatypes associados a um determinado datatypegroup. Associado ao requisito funcional **RF17**.
- **GET /datatypegroup/groups** - Pedido que devolve todos os datatypegroups no sistema. Associado ao requisito funcional **RF17**.
- **GET /threshold/groupthresholds** - Pedido que, para um dado datatypegroup, devolve todas as thresholds associadas a si. Associado aos requisitos funcionais **RF09** e **RF18**.
- **POST /customthreshold/customthresholds** - Pedido efetuado para adicionar as thresholds customizáveis a um determinado paciente. Associada aos requisitos funcionais **RF09** e **RF18**.
- **GET /disease/diseases** - Pedido que devolve todas as doenças no sistema. Associada aos requisitos funcionais **RF04** e **RF15**.

Questionnaires

No caso do serviço *Questionnaires*, apenas se criaram dois *controllers*, um para o mapeamento dos pedidos associados aos questionários, e outro para o mapeamento dos pedidos associados a doenças, sendo que para os *services* e *repositories* se optou pela mesma estratégia da utilizada no serviço *measurements*, isto é, uma para cada entidade. A pasta *DTOs* possui um único ficheiro, utilizado para decodificar o *token* que vem associado aos pedidos do utilizador, e assim obter o seu identificador, e outras informações caso necessário. Por sua vez, a pasta *Enums* possui os enumerados criados, nomeadamente para o tipo de notificações a enviar, e o estado da sessão associado a um questionário. As pastas *Config* e *Schemas* e o ficheiro *application.properties* seguem a mesma estratégia e estrutura à do serviço anterior. Na Figura 23 está representada a estruturação do serviço *Questionnaires*.

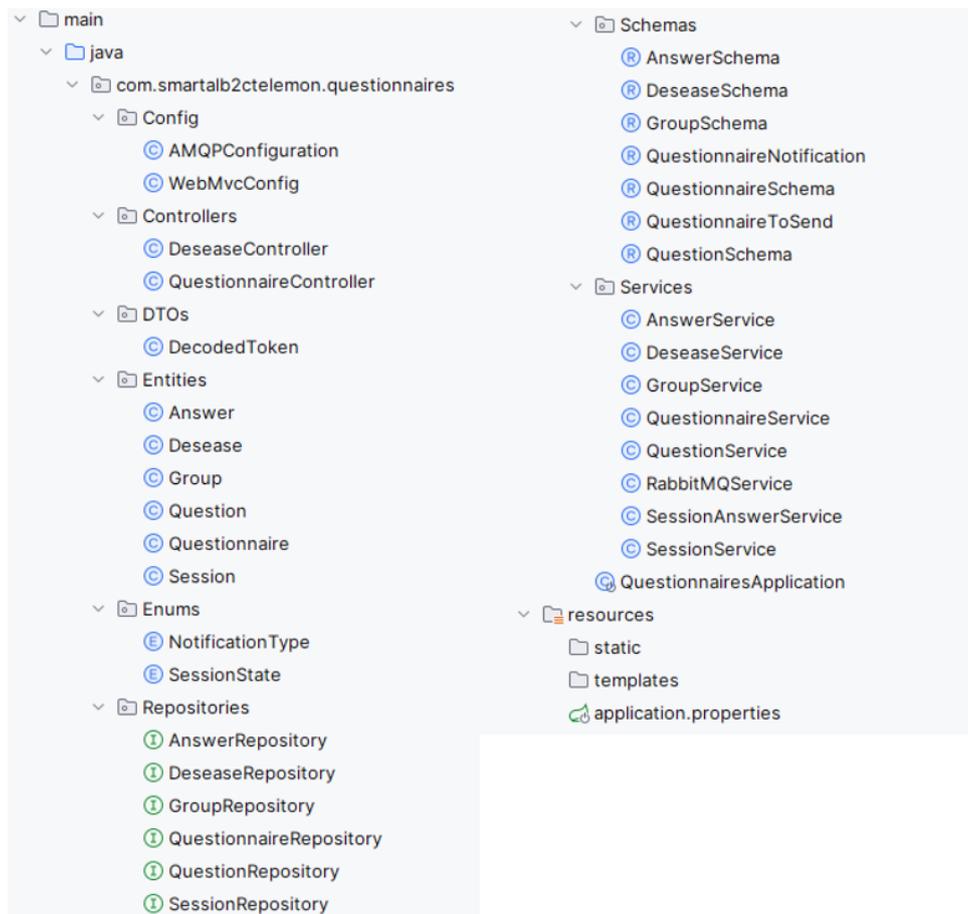


Figura 23: Implementação do Serviço Questionnaires

Na Figura 24 está representada a estrutura do *QuestionnaireController*, que necessita de comunicar com todas as classes *service*, e assim conseguir processar todos os pedidos associados aos questionários, efetuados pelos utilizadores.

A cada serviço é associada uma rota. Uma dessas rotas está representada na Figura 25 e representa a função responsável por criar uma nova sessão de resposta a um questionário. Analisando a Figura 25, o primeiro passo passa por obter o identificador do paciente, através do *token* de autorização enviado no pedido. Posteriormente, define-se a data de criação para a atual e, como está definido na linha 134, obtém-se o questionário cujo identificador corresponde àquele que foi enviado como parâmetro no pedido. Uma vez obtida toda a informação necessária, cria-se a nova sessão no sistema.

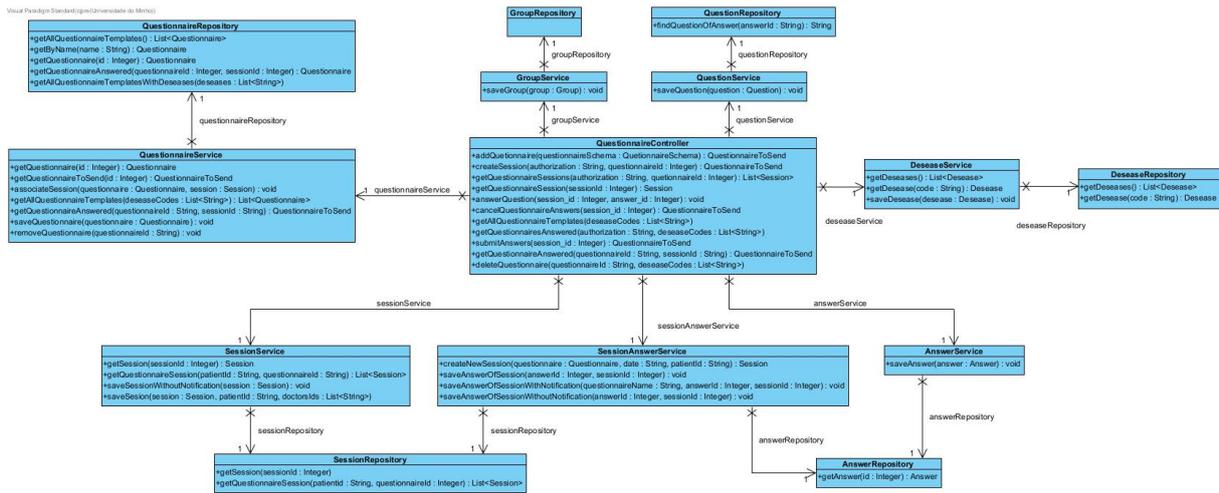


Figura 24: Estrutura QuestionnaireController

```

124     @PostMapping("@v/createSession")
125     public ResponseEntity<QuestionnaireToSend> createSession(
126         @RequestHeader("Authorization") String authorization,
127         @RequestParam("questionnaire_id") Integer questionnaireId) {
128         DecodedToken decode = new DecodedToken();
129         String patientId = decode.getSub(authorization);
130
131         String date = ZonedDateTime.now(
132             ZoneId.of(ZoneId.of("Europe/London"))
133         ).format(DateTimeFormatter.ofPattern("uuuu-MM-dd HH:mm"));
134
135         Questionnaire questionnaire = this.questionnaireService.getQuestionnaire(questionnaireId);
136         Session session = this.sessionAnswerService.createNewSession(questionnaire, date, patientId);
137         session.setSessionState(SessionState.NEW);
138         this.questionnaireService.associateSession(questionnaire, session);
139
140         return ResponseEntity.status(HttpStatus.OK).body(this.questionnaireService.getQuestionnaireAnswered(questionnaireId.toString(), session.getId().toString()));
141     }

```

Figura 25: Endpoint para criar nova sessão

Para além do exemplo acima, foram definidas mais rotas, que serão apresentadas de seguida, juntamente com os requisitos funcionais associadas a cada uma:

- **POST /questionnaire/savequestionnaire** - Pedido efetuado para guardar um determinado questionário. Associada ao requisito funcional **RF19**.
- **POST /questionnaire/createsession** - Pedido efetuado para criar uma nova sessão de resposta a um questionário. Associado ao requisito funcional **RF12**.
- **GET /questionnaire/questionnairesessions** - Pedido que devolve todas as sessões criadas por um paciente, associadas a um determinado questionário. Associado aos requisitos funcionais **RF12**, **RF13** e **RF14**.
- **GET /questionnaire/questionnairesession** - Pedido que devolve uma determinada sessão pretendida. Associado ao requisito funcional **RF13**.

- **POST /questionnaire/answerquestion** - Pedido que armazena a resposta dada a uma pergunta, uma determinada sessão. Associado ao requisito funcional **RF12**.
- **POST /questionnaire/cancelquestionnaireanswers** - Pedido efetuado para sinalizar que um paciente saiu a meio da resposta a um questionário. Associado ao requisito funcional **RF14**.
- **GET /questionnaire/allquestionnairetemplates** - Pedido efetuado de modo a obter todos os questionários existentes no sistema. Associado aos requisitos funcionais **RF12**, **RF13** e **RF14**.
- **GET /questionnaire/questionnairesanswered** - Pedido efetuado para obter todos os questionários aos quais um paciente já respondeu. Associado aos requisitos funcionais **RF12**, **RF13** e **RF14**.
- **POST /questionnaire/submitanswers** - Pedido efetuado para informar que uma sessão foi completada, e portanto deve ser considerada como submetida. Associado ao requisito funcional **RF12**.
- **GET /questionnaire/questionnaireanswered** - Pedido que devolve um questionário que possui uma determinada sessão. Associado aos requisitos funcionais **RF12**, **RF13** e **RF14**.
- **POST /questionnaire/deletequestionnaire** - Pedido elimina um determinado questionário do sistema. Associado ao requisito funcional **RF20**.

Shared Info

O serviço *Shared Info* segue a mesma estrutura dos serviços abordados anteriormente, com a exceção de não possuir um ficheiro de configuração para as notificações, uma vez que este serviço não necessita de enviar nenhuma notificação aos utilizadores. Uma vez que se trata de um serviço relativamente pequeno, apenas se criou um único *controller*, denominado *PatientDoctorController*. Na Figura 26 está representada a estruturação do serviço *Shared Info*.

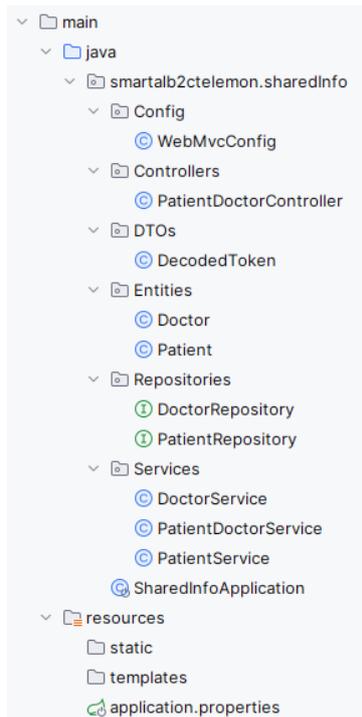


Figura 26: Implementação do Serviço Shared Info

À semelhança dos outros serviços, também se irá recorrer a um exemplo para perceber a estrutura do serviço *Shared Info*. O exemplo a considerar está representado na Figura 27, onde se apresenta a estrutura do *PatientDoctorController*. Através da análise da Figura, conclui-se que este serviço é de dimensão menor, comparativamente com os restantes serviços abordados previamente, uma vez que apenas tem de armazenar os identificadores dos pacientes e médicos que partilham informações.

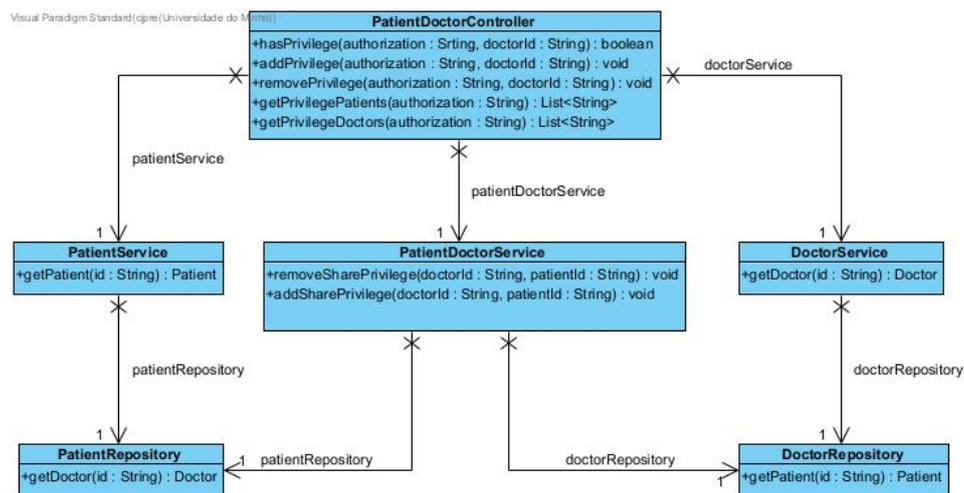


Figura 27: Estrutura PatientDoctorController

Uma dos *endpoints* definidos no serviço trata de verificar se um determinado paciente concedeu privilégios de partilha a um determinado médico. Como se pode comprovar na Figura 28, a função recebe o *token* de autorização, de onde se extrai o identificador do paciente em questão, e recebe também como parâmetro, o identificador do médico que se pretende verificar se tem ou não privilégios. Com base nos dois identificadores obtidos, apenas basta obter o paciente associado ao identificador fornecido e verificar se na lista dos seus médicos existe o identificador do médico recebido, tal como se pode comprovar na linha 46.

```
38     @GetMapping(path = "/hasprivilege")
39     public boolean hasPrivilege(
40         @RequestHeader("Authorization") String authorization,
41         @RequestParam(value = "doctorId") String doctorId
42     ) {
43         DecodedToken decode = new DecodedToken();
44         String patientId = decode.getSub(authorization);
45         Optional<Patient> optionalPatient = this.patientService.getPatient(patientId);
46
47         return optionalPatient.map(patient -> patient.getDoctors().stream().map(Doctor::getId).toList().contains(doctorId)).orElse(false);
48     }
49 }
```

Figura 28: Endpoint para verificar privilégios

Para além do exemplo acima referido, foram definidas as seguintes rotas:

- **POST /sharedinfo/addprivilege** - Pedido que adiciona privilégios de partilha de um paciente a um determinado médico. Associado ao requisito funcional **RF10**.
- **POST /sharedinfo/removeprivilege** - Pedido que remove privilégios de partilha de um paciente a um determinado médico. Associado ao requisito funcional **RF11**.
- **GET /sharedinfo/privilegepatients** - Pedido que devolve os identificadores dos pacientes que concederam privilégios de partilha a um determinado médico. Associado ao requisito funcional **RF15**.
- **GET /sharedinfo/privilegedoctors** - Pedido que devolve os identificadores dos médicos que têm privilégios de partilha com um determinado paciente. Associado ao requisito funcional **RF15**.

4.2.2 Frontend

A *frontend* corresponde à parte de aplicação com a qual o utilizador irá interagir, sendo responsável por apresentar todas as interfaces e comunicar com a *backend*, para assim apresentar a informação pretendida aos utilizadores.

Tal como foi abordado no capítulo das tecnologias, a presente aplicação foi desenvolvida utilizando a biblioteca *React*. Esta biblioteca é baseada na utilização de componentes, tendo cada um a sua própria lógica e estado, o que permite assim criar páginas onde toda a lógica está definida no seu próprio componente. Na figura 29 está representada a organização da *frontend* do sistema.

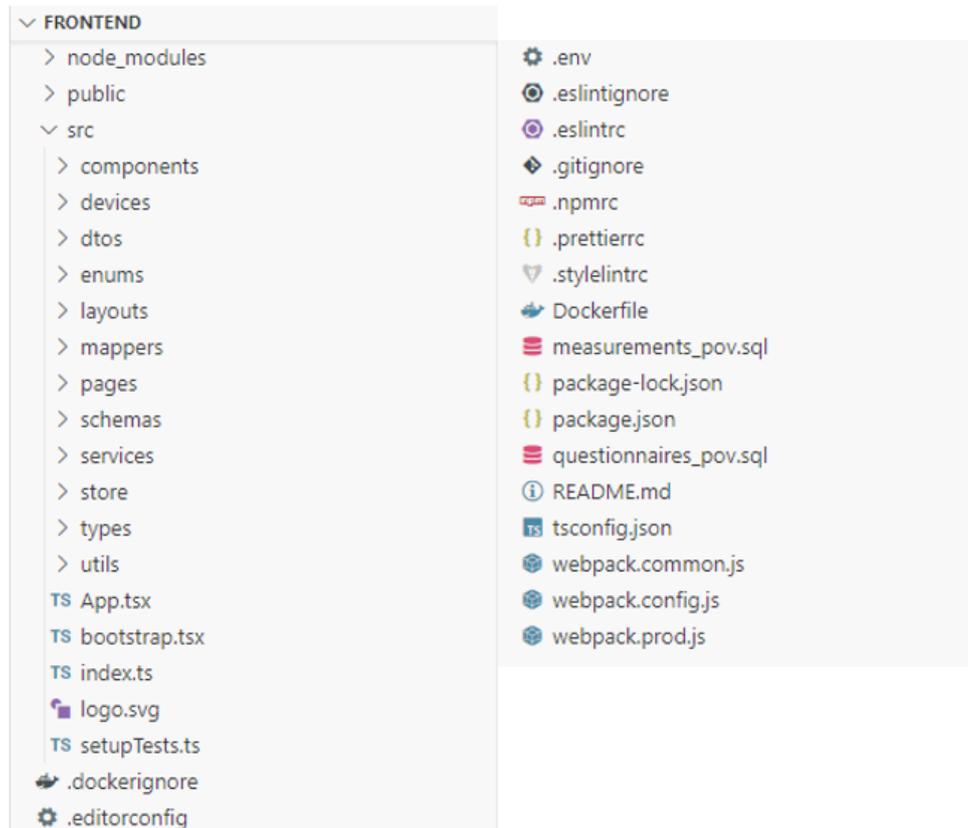


Figura 29: Estrutura Frontend

Para manter uma boa estrutura e homogeneidade entre os ficheiros, utilizou-se o formatador de código *prettier* e o analisador de código *eslint*. Os restantes constituintes do sistema podem ser explicados da seguinte forma:

- **public** - Pasta que contém imagens e ficheiros que podem ser acedidos em qualquer parte da aplicação.
- **components** - Possui todos os componentes desenvolvidos e que são utilizados para a construção das diferentes *pages*. De modo a manter a organização da aplicação, cada componente é constituído por três ficheiros distintos, um para a construção do componente, um para a definição da lógica, desde definição de variáveis até a chamada a **APIs**, e outro para a definição dos estilos a utilizar no componente.

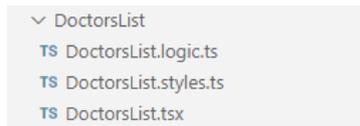


Figura 30: Estrutura Components

Na Figura 30 está demonstrada a típica estrutura de um componente, onde se team a lógica definida no ficheiro `.logic.ts`, os estilos definidos no ficheiro `.styles.ts`, e o componente a utilizar definido no ficheiro `.tsx`.

- **devices** - Pasta que possui toda a lógica associada aos aparelhos *Bluetooth*, desde configuração destes, até à maneira como os dados obtidos deste são tratados. Sempre que for necessário adicionar compatibilidade com um determinado aparelho, apenas será necessário acrescentar o ficheiro de configuração desse mesmo aparelho.
- **dtos** - Pasta com todos os objetos onde se agrupa um conjunto de atributos, de forma a otimizar a comunicação
- **enums** - Possui todas as constantes associadas a um determinado tipo, e que assim facilitam a distinção de diferentes tipos na mesma classe
- **layouts** - Uma vez que há dois tipos de utilizadores, foi necessário definir um *layout* para cada um deles. A intenção dos *layouts* é apenas dar acesso a certas partes da aplicação a utilizadores que tenham permissão para aceder a essas. Caso contrário, serão redirecionados para um página a informar do acesso negado.
- **mappers** - Toda a informação proveniente das **APIs** vem sob a forma de *schemas*, e de modo a conseguir transformar esse tipo de dados em *DTOs* que podem ser utilizados na aplicação são utilizados *mappers*, uma para cada tipo de dados.
- **pages** - Possui as páginas que constituem a aplicação.
- **schemas** - Contém as estruturas que definem o tipo de dados proveniente das chamadas às **API**.
- **services** - Para cada serviço da aplicação, definiu-se um ficheiro `.service.ts`, tal como se pode observar na figura 31. Cada um dos ficheiros possui código que permite efetuar chamadas a cada uma das **API** e devolver, caso possível, o resultado obtido.

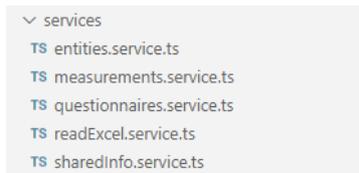


Figura 31: Estrutura Services

- **store** - Quando é necessário definir variáveis que devem ser acedidas em qualquer parte da aplicação, cria-se um ficheiro context.tsx, evitando assim ter de passar a informação relevante como *props* em diferentes componentes.
- **types** - Pasta onde estão definidos alguns dos tipos de dados a utilizar na aplicação.
- **Dockerfile** - Uma vez que a aplicação é colocada a executar em ambiente *docker*, é necessário criar um *dockerfile*, onde se define como é criada a imagem da aplicação.

Como tem vindo a ser referido ao longo do documento, um dos principais requisitos da aplicação é permitir a realização de medições através de dispositivos *Bluetooth*. Para tal, foi necessário desenvolver um conjunto de funções na *frontend*, que permitisse a conexão e leitura dos valores dos aparelhos, de forma o mais abstrata possível, ou seja, independentemente do aparelho em questão, as etapas associadas à medição sejam exatamente as mesmas.

Na Figura 32 está representada a lista com os dispositivos utilizados. A *key* de cada lista representa o nome do aparelho, que é necessária para identificar quais as diferentes propriedades que se têm de obter. A propriedade serviceUUID permite aceder ao serviço de transmissão dos aparelhos. Adicionalmente, a propriedade characteristicUUID indica qual o identificador da característica dentro do serviço à qual se tem de aceder para obter os valores de leitura. A propriedade devices permite saber qual o tipo de dispositivo em questão, tais como oxímetro, termómetro ou tensiómetro, e quais os restantes detalhes desse aparelho.

```

24 const devicesByName = Object.freeze({
25   BerryMed: {
26     characteristicUUID: '49535343-1e4d-4bd9-ba61-23c647249616',
27     devices: [
28       {
29         device: OximeterBerryMed,
30         type: DeviceType.OXIMETER
31       }
32     ],
33     serviceUUID: '49535343-fe7d-4ae5-8fa9-9fafd205e455'
34   },
35   BluetoothBP: {
36     characteristicUUID: '0000fff1-0000-1000-8000-00805f9b34fb',
37     devices: [
38       {
39         device: TermometerBluetoothBP,
40         type: DeviceType.TERMOMETER
41       },
42       {
43         device: TensiometerBluetoothBP,
44         type: DeviceType.TENSIOMETER
45       }
46     ],
47     serviceUUID: '0000fff0-0000-1000-8000-00805f9b34fb'
48   }
49 });

```

Figura 32: Lista de Propriedades dos Dispositivos

```

9   export const OximeterBerryMed = {
10     eventHandler: handleChangeCharacteristicOximeterBerryMed,
11     measureGroups: [DataTypeGroup.SPO, DataTypeGroup.HBR]
12   };
13
14   export const TermometerBluetoothBP = {
15     eventHandler: handleChangeCharacteristicTermometerBluetoothBP,
16     measureGroups: [DataTypeGroup.TMP]
17   };
18
19   export const TensiometerBluetoothBP = {
20     eventHandler: handleChangeCharacteristicTensiometerBluetoothBP,
21     measureGroups: [DataTypeGroup.BLP, DataTypeGroup.HBR]
22   };

```

Figura 33: Dispositivos Existentes

Dependendo do aparelho em questão, os valores serão tratados de maneira diferente. Como se pode observar na Figura 33, para cada aparelho, está definido um *eventHandler*, uma função com o objetivo de validar e tratar os valores a serem lidos, enquanto que a propriedade *measureGroups* indica quais os diferentes grupos que estão a ser recolhidos, tais como temperatura ou pressão sanguínea.

```

80  const eventHandler = (
81  devices: any[],
82  setError: React.Dispatch<React.SetStateAction<string | undefined>>,
83  setStructure: React.Dispatch<
84  React.SetStateAction<
85  {
86    comment: string;
87    types: {
88      type: MeasurementSchemaDataType;
89      unit: string;
90    }[];
91  }[]
92  >
93  >
94  ) => {
95  return (event: any) => {
96  if (event.target.value.buffer) {
97  const currentValues = new Uint8Array(event.target.value.buffer);
98  const deviceType = deviceCheck(currentValues);
99  const device = devices.filter((device) => device.type === deviceType)[0]
100 | .device;
101  const deviceEventHandler = device.eventHandler;
102
103  if (!readingInfo.readingStarted) {
104  const groups = device.measureGroups;
105
106  groups.forEach((group: DataTypeGroup) => {
107  getGroupDataTypes(group).then((response) => {
108  setStructure((prev) => [...prev, response]);
109  });
110  });
111  }
112
113  deviceEventHandler(currentValues, dispatch, setError);
114  }
115  });
116  };

```

Figura 34: Função EventHandler

Tal como foi referido, o *eventHandler* trata os dados recebidos. Na Figura 34 está representada a implementação desse mesmo. Analisando a Figura, observa-se que os valores a serem recebidos são armazenados na variável *currentValues* e com base nesses valores, determina-se qual o aparelho que está a transmitir esses valores. Posteriormente, obtém-se a função que trata os valores desse tipo de aparelho, armazenada na variável *deviceEventHandler* e chama-se essa mesma, para tratar desses valores.

Uma vez analisadas as funções anteriores, é possível prosseguir para a principal função responsável pelo emparelhamento com os dispositivos *Bluetooth*, que por sua vez utiliza as funções até agora mencionadas. Na Figura 35 está representada a função de conexão a um aparelho *Bluetooth*. Antes de se iniciar a conexão propriamente dita, é necessário determinar qual o tipo de aparelho que se está a tentar associar ao sistema. Tal é conseguido através da análise do nome do aparelho, que assim permite obter as características acima abordadas e necessárias para a conexão.

De seguida, o sistema conecta-se ao dispositivo e obtém a característica pretendida. Uma vez conectado, a função fica à espera que os valores sejam transmitidos, sendo que à medida que os dados estão a ser transmitidos, a função *eventHandler* abordada acima irá tratar desses mesmos dados.

```

118 const connect = (
119   device: BluetoothDevice,
120   setError: React.Dispatch<React.SetStateAction<string | undefined>>,
121   setStructure: React.Dispatch<
122     React.SetStateAction<
123       {
124         comment: string;
125         types: {
126           type: MeasurementSchemaDataType;
127           unit: string;
128         }[];
129       }[]
130     >
131   >
132 ) => {
133   const { characteristicUUID, devices, serviceUUID } = getDevice(
134     (device.name as any).replace(/\\s/g, '') as keyof typeof devicesByName
135   );
136
137   // Conectar ao aparelho
138   device.gatt
139     ?.connect()
140     .then(
141       // Obter o serviço pretendido
142       (server) => {
143         return server?.getPrimaryService(serviceUUID);
144       }
145     )
146     .then(
147       // Obter características pretendidas
148       (service) => {
149         return service?.getCharacteristic(characteristicUUID);
150       }
151     )
152     .then((characteristic) => {
153       return characteristic?.startNotifications();
154     })
155     .then((value) => {
156       changeCanStart(true);
157       value?.addEventListener(
158         'characteristicvaluechanged',
159         eventHandler(devices, setError, setStructure)
160       );
161     })
162     .catch((err) => {
163       console.error(err);
164     });
165 };

```

Figura 35: Função connect

4.3 Interfaces

Tendo toda a lógica definida e explicada, é possível prosseguir para a demonstração da aplicação, nomeadamente das suas interfaces a as funcionalidades associadas a cada uma delas. Como há dois tipos de utilizadores, os médicos e os pacientes, irão ser distinguidas as interfaces para cada tipo. Nesta primeira secção serão abordados as interfaces que não dependem do tipo de utilizador, e só depois se irá falar das interfaces de cada tipo.

Tal como em qualquer aplicação, independentemente do tipo de utilizador, é necessária uma página de autenticação no sistema. Como se pode observar na figura 36, sempre que um utilizador acede ao domínio da aplicação, terá de introduzir as suas credenciais. Dependendo do tipo de utilizador, este será reencaminhado ou para a página dos paciente, ou para a página dos médicos.

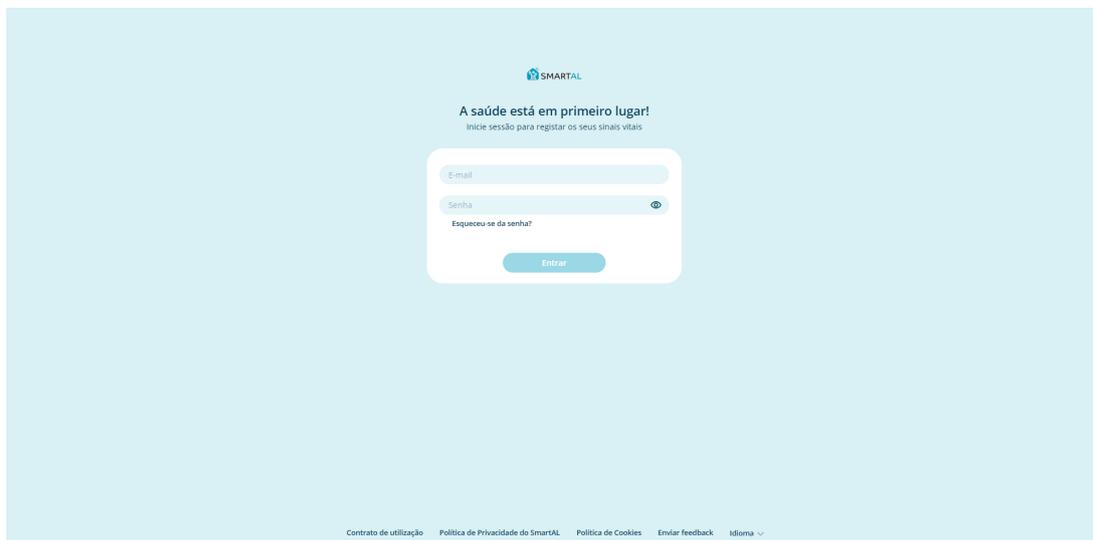


Figura 36: Login

Adicionalmente, e já autenticado, o utilizador pode alterar as suas informações. Como se pode observar na Figura 37, na *tab* de informações de conta pode alterar desde nome, número de telemóvel, entre outros, e na *tab* de segurança pode alterar a sua *password*.



Figura 37: Editar Informação

Sempre que um paciente efetuar uma medição ou preencher um questionário, quer ele quer cada um dos seus médicos irão ser notificados acerca de tal ação. Estas notificações permitem, especialmente aos médicos, estar a par das ações dos seus pacientes. Na Figura 38, no canto superior direito, estão representadas as notificações na *inbox* do paciente, associadas à realização de três novas medições.

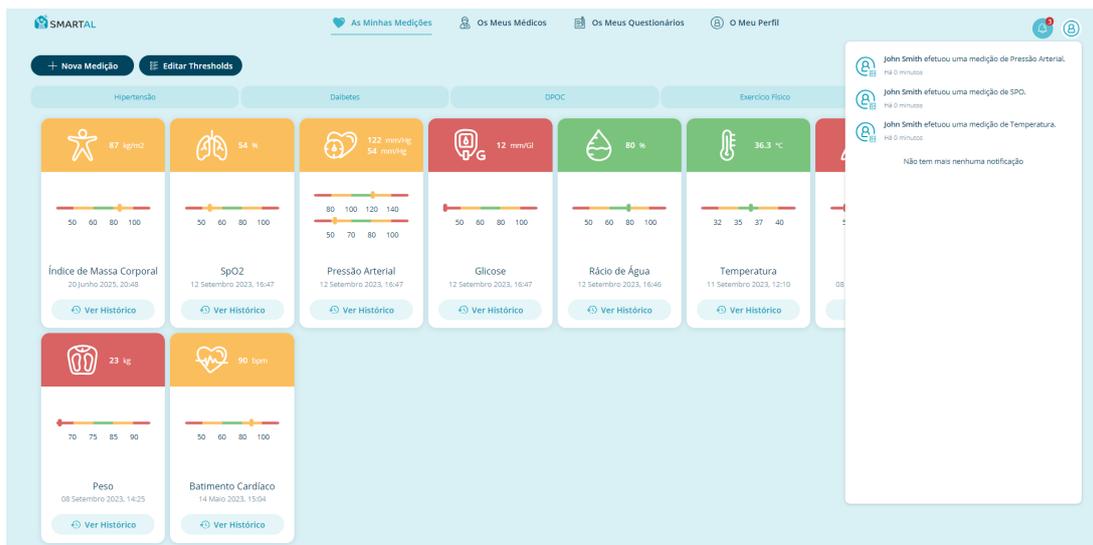


Figura 38: Notificações

Por fim, e não menos importante, foi necessário fornecer algum tipo de *feedback* aos utilizadores, informando acerca do sucesso ou insucesso das suas ações. Para tal, foram desenvolvidos dois componentes, um para os casos de sucesso, representado a verde, e outro para os casos de erro, representado a vermelho. Esta distinção por cores permite aos utilizadores saberem imediatamente se a ação foi concluída. Na figura 39, é possível ver a vermelho um caso onde um médico não conseguiu adicionar um questionário no sistema, e a verde um caso onde um paciente adicionou com sucesso privilégios de partilha a um médico.

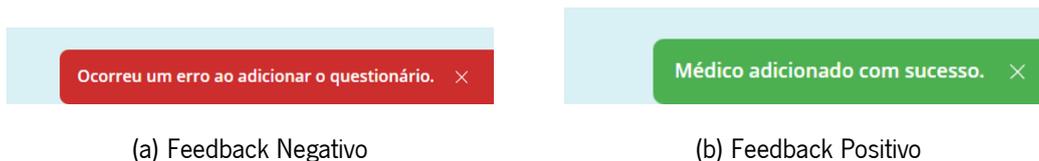


Figura 39: Feedback

4.3.1 Paciente

As Minhas Medições

Uma vez autenticado na aplicação, a primeira página apresentada será a das medições do próprio paciente, caso existam. Como se pode observar na Figura 40, a página é composta por três secções distintas, a primeira, representada por um tom mais escuro, inclui as principais ações que o utilizador pode efetuar, sendo estas efetuar uma nova medição, editar as suas *thresholds* e gerar um *PDF* com as medições pretendidas. A segunda secção inclui as doenças pelas quais o paciente pode filtrar as medições, e a terceira inclui as medições propriamente ditas.

Cada cartão informa acerca do nome da medição, a altura em que foi realizada, e por fim, como se compara o valor medido com os *thresholds* definidos. De modo a facilitar a leitura de valores, optou-se por utilizar cores que simbolizassem o estado do valor, sendo que verde representa um bom valor, o amarelo um valor intermédio, e o vermelho um sinal de alerta.

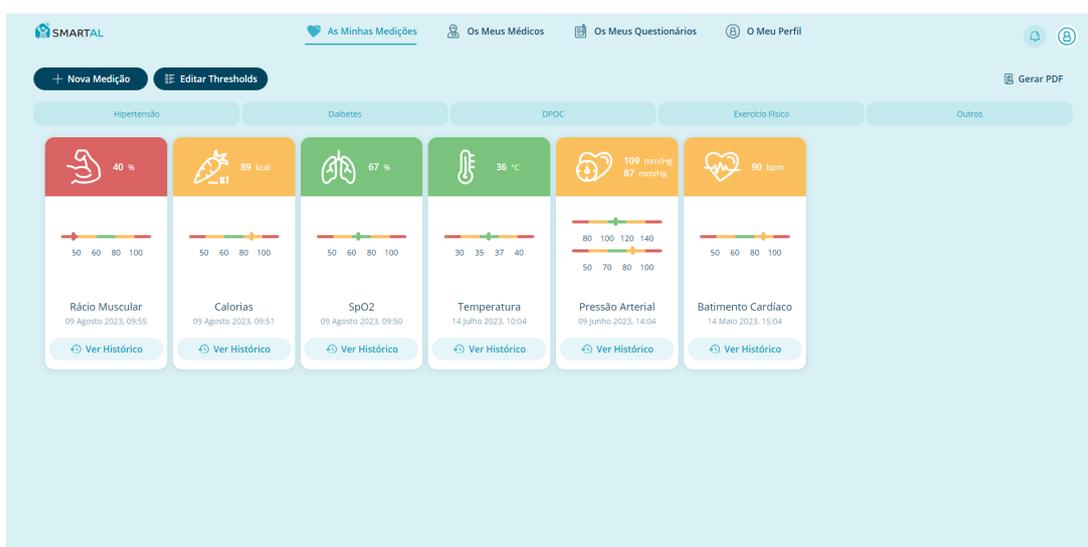
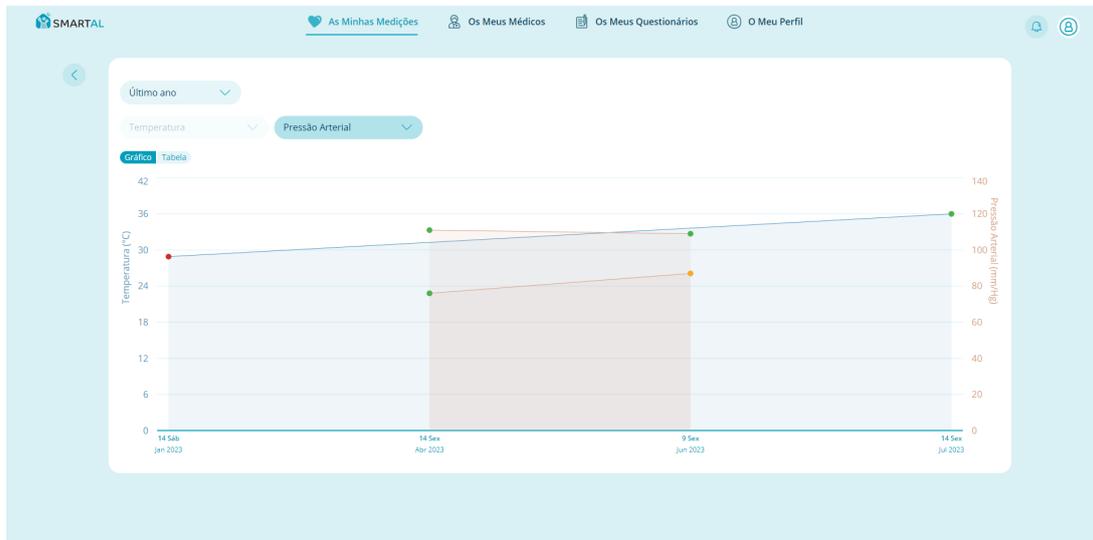


Figura 40: Página das Medições

Cada cartão possui um botão de visualização de histórico, que, quando pressionado, redireciona

o paciente para uma página com um gráfico onde estarão dispostos os valores de todas as medições efetuadas, associadas ao tipo de medida pretendida. Caso pretenda, pode também escolher um segundo grupo de medições para visualizar simultaneamente. Na eventualidade da disposição do gráfico parecer confusa, o utilizador poderá optar por ver os valores sob a forma de tabela, tal como se pode observar na Figura 41.



(a) Modo Gráfico

Data	Manual	Quem Realizou	Medição	Estado	Valores
14 Julho 2023, 10:04	×	JSmith	Temperatura	✓	36 °C
14 Janeiro 2023, 11:04	×	JSmith	Temperatura	✗	28.9 °C
09 Junho 2023, 14:04	×	JSmith	Pressão Arterial	⚠	109/87 mm/Hg
14 Abril 2023, 15:04	×	JSmith	Pressão Arterial	✓	111/76 mm/Hg

(b) Modo Tabela

Figura 41: Página com Histórico

Selecionando a opção de adicionar uma nova medição, irá aparecer um modal com as opções de tipo de leitura que o utilizador pode efetuar, sendo estas modo *Bluetooth* ou modo manual. No modo manual, o utilizador terá de, numa primeira fase, selecionar qual a medição que irá realizar e após tal definido, irão aparecer os campos referentes aos diferentes valores que este tem que introduzir. Por fim,

é possível deixar um comentário, caso tenha existido alguma situação importante a apontar ao realizar a medição.

No caso da leitura *Bluetooth*, o utilizador apenas terá de ligar o aparelho que irá utilizar, conectar com a aplicação, e quando estiver preparado começar a efetuar a medição. Quando esta terminar, os valores obtidos irão ser apresentados no ecrã. Há semelhança da leitura manual, o utilizador também poderá deixar um comentário caso sinta que seja necessário.

Outra funcionalidade descrita foi a edição das *thresholds*. Quando o utilizador seleciona essa opção, é redirecionado para uma página onde são apresentadas todos os tipos de medições que se podem realizar na aplicação, e para cada uma delas, os valores das suas *thresholds*. No caso do paciente atualizar as suas *thresholds*, tal mudança apenas será efetuada no seu perfil, permitindo assim que vários utilizadores possam ter diferentes *thresholds*.

Por fim, é possível criar um relatório *PDF* com a informação das medidas pretendidas. Para tal, o utilizador terá de selecionar quais os tipos de medidas que pretende guardar, e qual o período de tempo que se deve ter em conta, isto é, com base no período definido, irão apenas surgir as medições efetuadas nesse intervalo.

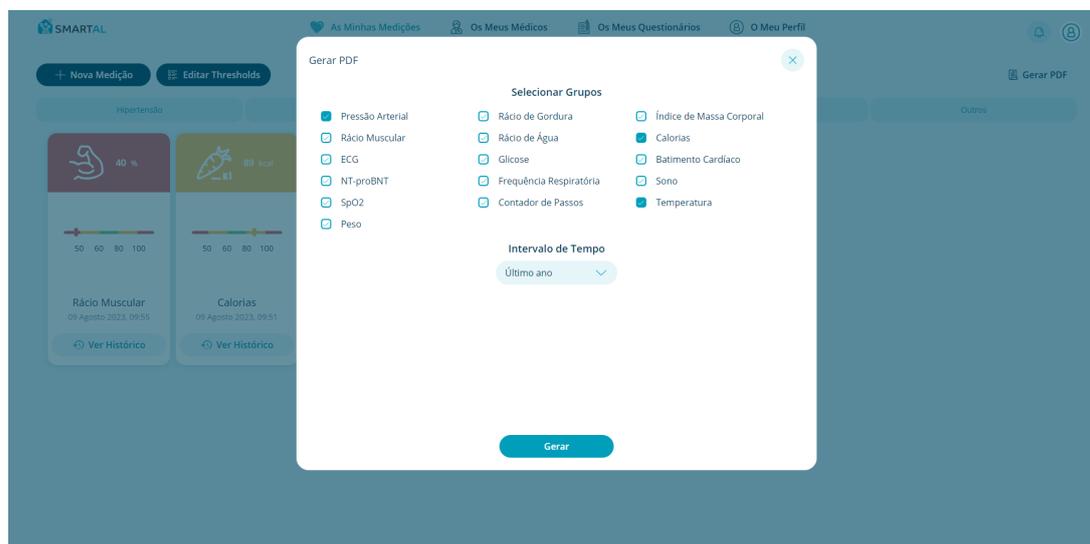
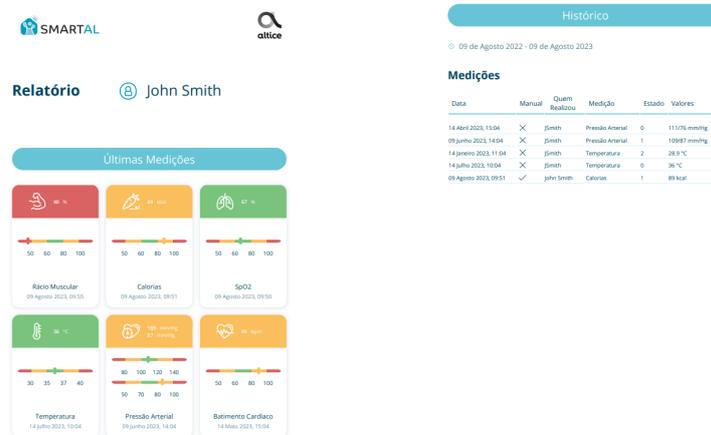


Figura 42: Página de Gerar PDF

O relatório gerado possui o histórico de todas as medições de cada tipo de medida pretendida. No histórico, tal como se pode verificar na Figura 43, são indicadas a data de realização da medição, se a medição foi realizada manualmente ou via *Bluetooth*, quem a realizou, o tipo de medição que foi realizada, o estado desta, sendo que 2 simboliza uma medição perigosa, 1 representa um valor intermédio, e 0 uma medição boa, bem como a indicação dos valores associados a cada medição. O paciente pode depois

guardar o documento e dar-lhe o devido uso, sem estar limitado pela aplicação.



(a) Parte Inicial

(b) Parte Final

Figura 43: Relatório Gerado

Os Meus Médicos

Relativamente aos seus médicos, é possível visualizar todos os médicos disponíveis na plataforma, agrupados pela sua especialidade. Todos os médicos com a *checkbox* preenchida correspondem aos médicos com os quais o paciente decidiu partilhar a sua informação. Caso um paciente pretenda remover ou adicionar privilégios de partilha a um médico, basta selecionar a *checkbox* e confirmar a sua decisão, tal como se verifica na Figura 44.

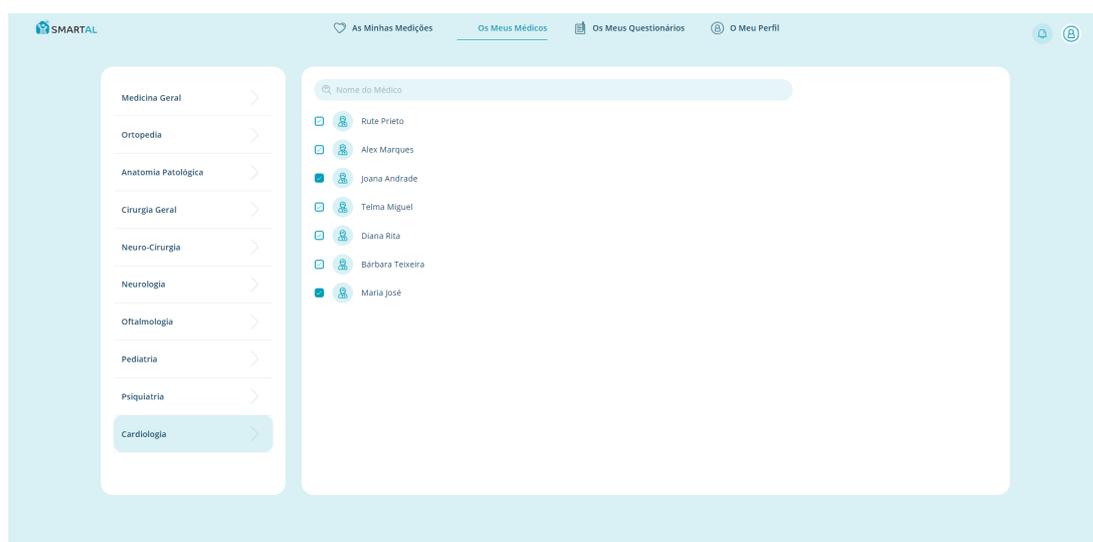


Figura 44: Os Meus Médicos

Os Meus Questionários

Na página dos questionários, tal como se pode verificar na 45, o paciente irá poder visualizar todos os questionários disponibilizados no sistema, podendo filtrar estes pelo tipo de doença associado a um questionário. Ao selecionar um questionário, irão ser apresentadas, caso existam, todas as tentativas que o paciente efetuou previamente, quer estas estejam concluídas ou não. Para as submissões em curso, poderá ver as respostas selecionadas até ao momento, pressionando o ícone com as três barras horizontais, ou então resumir a partir do estado em que o deixou, selecionando o lápis. Nos casos das submissões submetidas, apenas poderá visualizar as suas respostas.

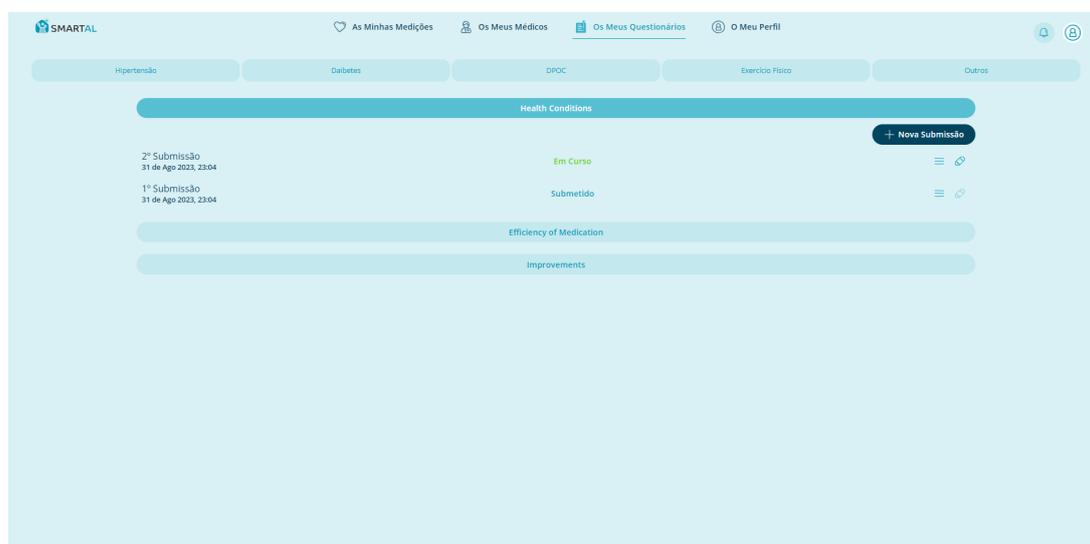


Figura 45: Questionários

Para iniciar uma nova medição, basta selecionar o botão de nova medição, sendo posteriormente redirecionado para a página de resposta a um questionário, tal como se pode observar na figura 46.

O Meu Perfil

Quando um paciente acede à sua área pessoal, poderá consultar informações como quais as suas alergias, medicações, hábito diários e informações mais simples como sexo, altura, peso e localização, tal como se pode observar na Figura 47.

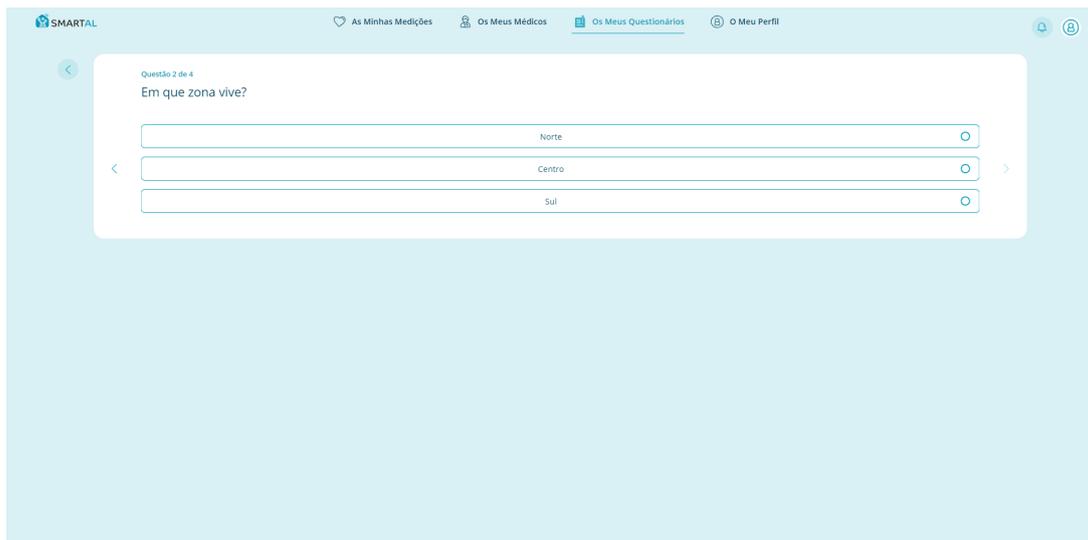


Figura 46: Responder Questionário

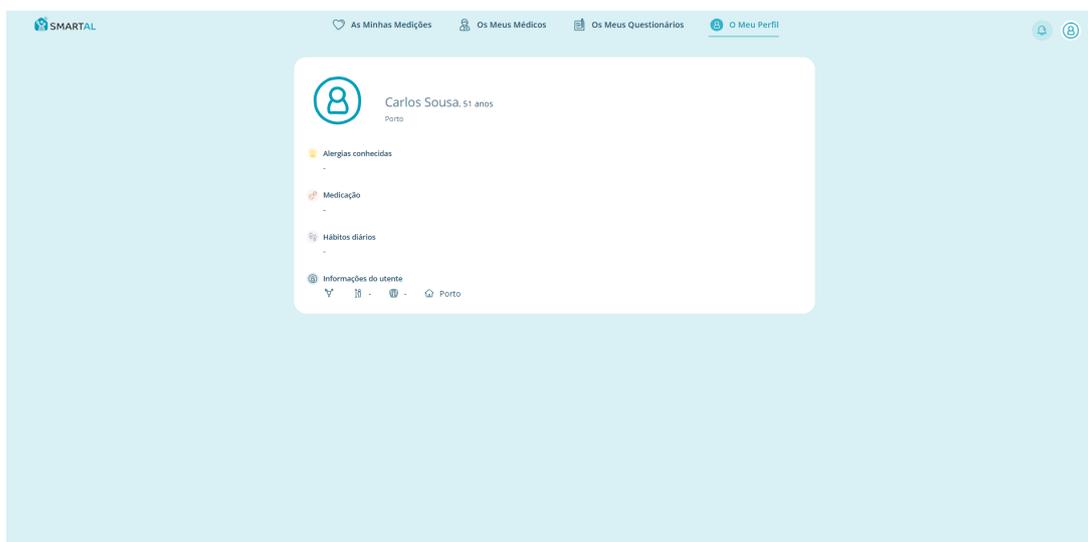


Figura 47: Perfil Paciente

4.3.2 Médico

Os Meus Pacientes

Na página dos seus pacientes, são apresentados todos os pacientes que decidiram partilhar informação com o médico em questão, podendo estes ser filtrados por nome. Cada paciente está representado num cartão, onde é possível visualizar não só um conjunto de informações, como também realizar algumas ações. Para além do avatar e do nome do paciente, existe uma secção, denominada de notificações,

onde estão representados o número de medições no estado de alerta e no número de medições no estado intermédio. Na última secção, está representada a última medição efetuada pelo paciente.

Em termos de ações, é possível ir visualizar o perfil do paciente, ou então visualizar com maior detalhe as informações do paciente. Uma vez que já se abordou a página de perfil, apenas se fará referência aos detalhes.

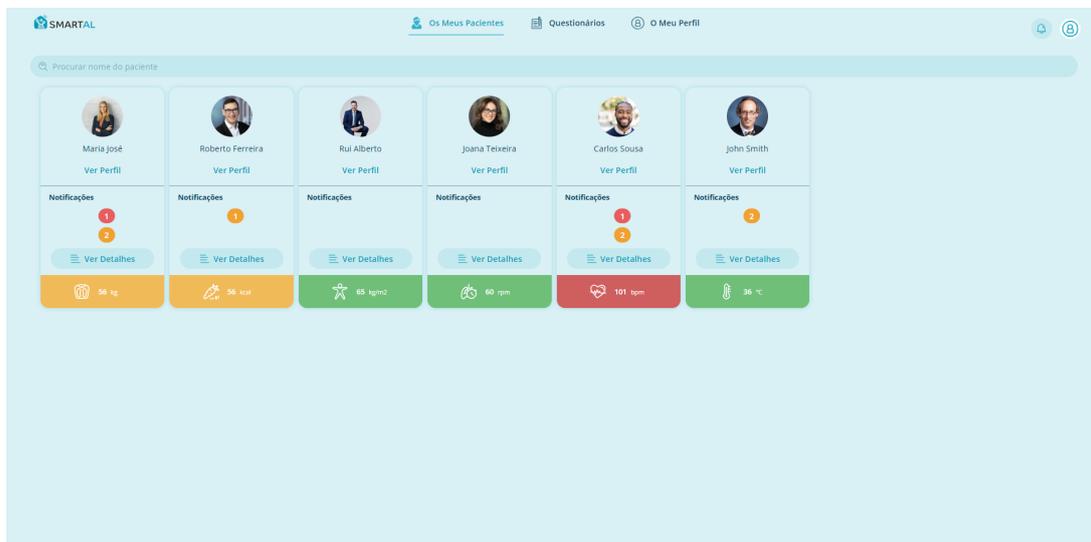


Figura 48: Os Meus Pacientes

Existem diversas informações a que um médico pode ter acesso, e por isso optou-se por listar as diferentes informações no topo da página. Como se pode ver na Figura 49, pode selecionar entre medições, *thresholds* e questionários. No que toca às vistas para cada uma destas opções, e para manter a consistência no sistema, a estrutura delas será semelhante às apresentadas anteriormente, na secção das interfaces do paciente, com a óbvia exceção de um médico não poder nem registar novas medições, nem responder a questionários.

Questionários

Uma das principais funções de um médico também passa por disponibilizar ou remover novos questionários no sistema. Na parte superior da página, está presente um componente que permite ao médico adicionar um ficheiro *excel* com a informação do questionário pretendido, e onde poderá posteriormente selecionar a doença a que o questionário acrescentar se irá referir.

Relativamente aos questionários já existentes, é possível verificar a composição de cada um, e caso se pretenda, eliminar estes.

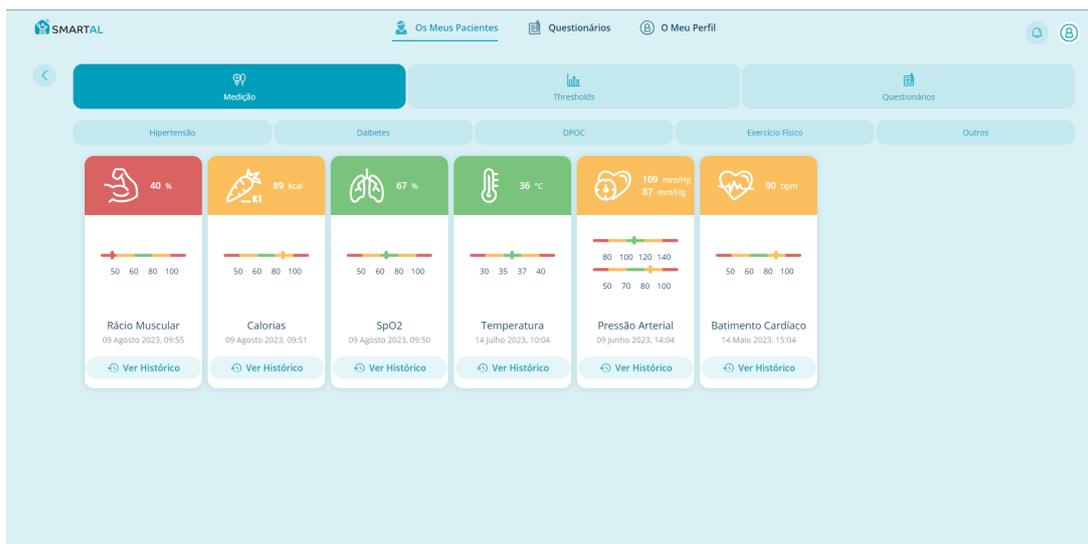


Figura 49: Detalhes de um Paciente

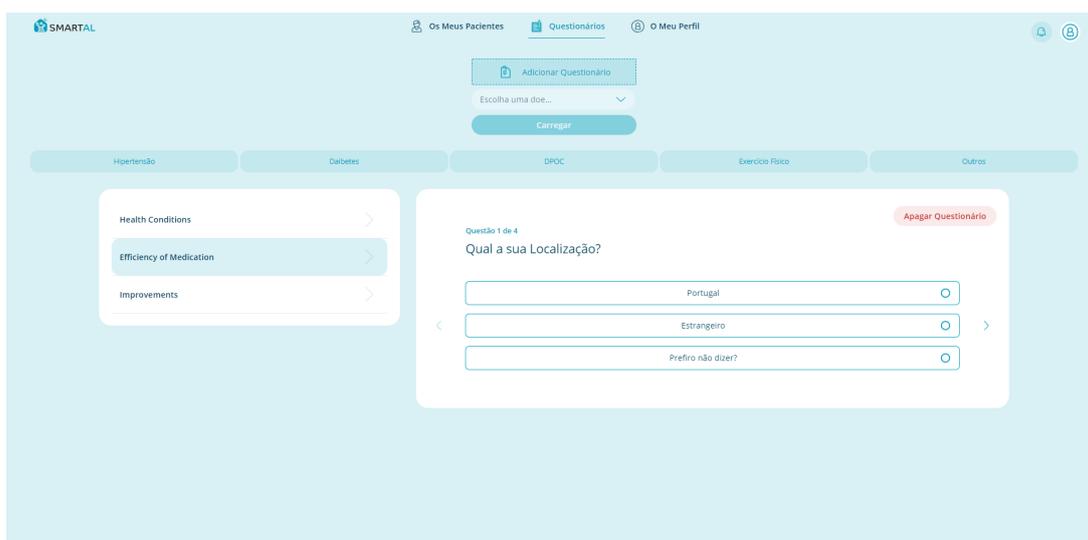


Figura 50: Questionários

O Meu Perfil

Na sua área pessoal, o médico pode visualizar as suas informações básicas, e ainda mais alguns detalhes, tais como idiomas, educação e experiência e seguros de saúde aceites. Tal informação encontra-se representada na Figura 51.

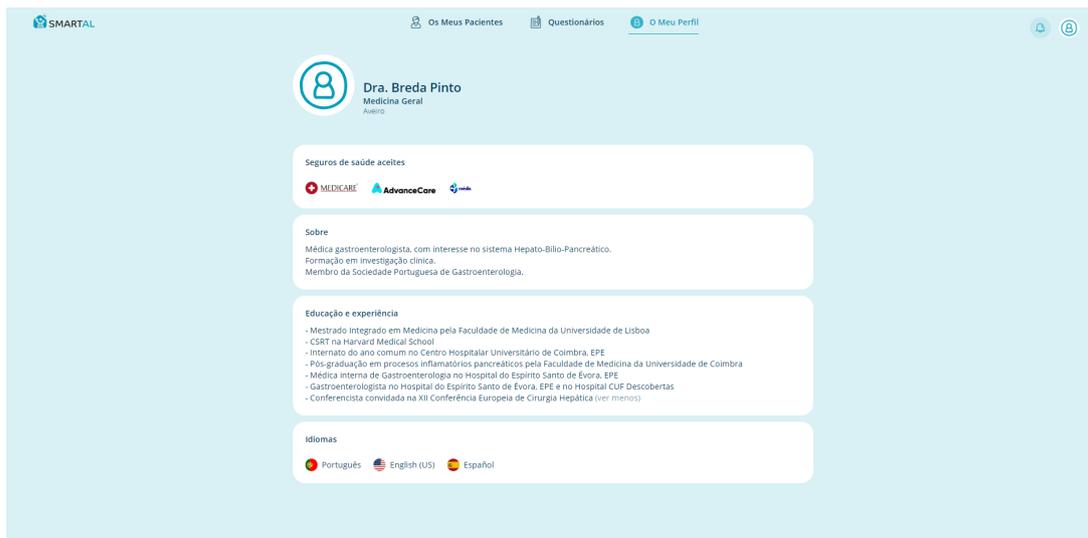


Figura 51: Perfil Médico

4.4 Dispositivos Utilizados

Uma vez que a aplicação suporta a utilização de aparelhos *Bluetooth*, foi necessário adquirir alguns equipamentos para assim validar o caso de estudo. Nesse contexto, foram utilizados três aparelhos distintos, um oxímetro, que permite efetuar a medição do pulso arterial e a saturação periférica de oxigénio, um tensiômetro, que também mede o pulso arterial, mas que o classifica de acordo com o seu nível e um termómetro, que permite a inferência da temperatura corporal através da medição sem contacto com o corpo.

Cada um destes aparelhos é constituído por determinados serviços e características, as quais é necessário descobrir e analisar, antes de se poder adicionar compatibilidade com o sistema. Para tal, obtive-se toda a informação que cada aparelho emite quando está a ser utilizado, e filtra-se por aquela que realmente importa para o contexto da aplicação, que são os valores da medição. No futuro, caso se pretenda adicionar compatibilidade com mais aparelhos, apenas será necessário saber qual o serviço e característica desse aparelho, verificar qual o tempo de leitura indicado e descobrir como tratar os valores obtidos, isto é, saber se os valores precisam ou não de tratamento antes de serem apresentados na aplicação ao cliente.



(a) Oxímetro



(b) Tensiômetro



(c) Termômetro

Figura 52: Dispositivos

4.5 Validação

Ao longo do desenvolvimento da aplicação foram realizadas reuniões semanais onde foi sucessivamente apresentado o estado da aplicação e se esta se enquadrava com o pretendido. Quando se verificou que os serviços estavam bem definidos, apenas foi necessário rever algumas decisões de *design* na *frontend* da aplicação, e ajustar esta caso necessário.

Uma vez concluído o projeto, foi necessário reunir com alguns elementos das equipas de *frontend* e *backend* da *Altice*, onde se reviram, uma vez mais, todas as funcionalidades implementadas e se verificou se os requisitos funcionais definidos estavam devidamente implementados.

Nas Figuras 53 e 54 estão representadas as validações efetuadas nos requisitos funcionais e não funcionais, respetivamente, e onde se identifica se os requisitos foram ou não cumpridos, e caso necessário, as observações efetuadas.

Requisito	Resultado	Observações
RF01	✓	-
RF02	✓	-
RF03	✓	-
RF04	✓	Utilização de um gráfico para visualizar o histórico das medições foi implementado como uma funcionalidade extra
RF05	✓	-
RF06	✓	-
RF07	✓	-
RF08	✓	-
RF09	✓	-
RF10	✓	-
RF11	✓	-
RF12	✓	-
RF13	✓	A possibilidade de filtrar a visualização das respostas com filtros foi implementado como um extra
RF14	✓	-
RF15	✓	-
RF16	✓	-
RF17	✓	-
RF18	✓	-
RF19	✓	-
RF20	✓	-

Figura 53: Validação Requisitos Funcionais

Requisito	Resultado	Observações
RNF01	✓	-
RNF02	✓	-
RNF03	✓	Como trabalho futuro pode-se implementar também as validações do lado do servidor
RNF04	✓	-
RNF05	✓	-
RNF06	✓	A validação foi limitada aos dispositivos existentes (oxímetro, termómetro, tensiómetro e glicómetro)
RNF07	✓	Os serviços guardam apenas os identificadores dos médicos e dos utentes, ficando a informação complementar guardada no serviço <i>Entities</i> .

Figura 54: Validação Requisitos Não Funcionais

Analisando as Figuras, percebe-se que todos o requisitos foram cumpridos com sucesso, e portanto, considera-se que o sistema ficou validado.

4.6 Deployment

A última etapa associada ao desenvolvimento do sistema consistiu em colocá-lo a correr em ambiente *cloud*. Para tal, recorreu-se à utilização do *kubernetes*, uma solução frequentemente utilizada não só para fazer *deployment*, como também escalar e gerir os serviços em *containers*. Em termos de *cloud*, optou-se por escolher a *Google Cloud*, uma vez que este é o *provider* mais utilizado para os projetos na *Altice Labs*.

No *deployment* foi necessário criar 4 máquinas virtuais, 1 para a *frontend* e 3 para os serviços de *backend*, e definir as características de cada uma das máquinas a instanciar. Após alguma discussão e testes de carga, optou-se por um total de 48 vCPU e 48 GB de RAM.

Capítulo 5

Conclusões e trabalho futuro

Ao longo deste capítulo será feita uma reflexão acerca do trabalho realizado e de como este se alinha com os objetivos pretendidos e, adicionalmente, serão discutidos alguns aspetos a ser melhorados, bem como algumas funcionalidades que, apesar de não fazerem parte dos requisitos da aplicação, poderiam ser implementados e melhorar ainda mais a aplicação.

5.1 Conclusões

O presente trabalho teve como objetivo o desenvolvimento de uma aplicação de telemonitorização, que permitisse aos pacientes efetuar as suas medições via *Bluetooth* e partilhar essa mesma informação com os seus médicos.

Numa fase inicial, foi efetuado um estudo acerca da área da *eHealth*, e exploradas algumas das ferramentas/tecnologias utilizadas nos últimos anos. Tal estudo passou por compreender o que é um serviço de telemonitorização, qual a sua estrutura típica e quais as principais doenças associadas a tais serviços. Uma vez que pretendia utilizar dispositivos *Bluetooth*, também foi necessário fazer um levantamento de alguns dos aparelhos frequentemente usados para tal tipo de medições, bem como entender como funciona a partilha de dados via *Bluetooth*.

De seguida, exploraram-se os pontos fortes e limitações tipicamente associadas a este tipo de serviços, de forma a tentar eliminar ou mitigar o impacto destes na aplicação. Tal exploração foi seguida da referência a alguns exemplos de casos de aplicação, quer nacionais como internacionais, e que ajudaram a ter não só um ponto de referência, como também saber quais os pontos onde poderia haver hipótese de melhorar.

Antes de se avançar para a parte prática, procedeu-se a uma análise do estado da prática na *Altice*, para saber que tipos de tecnologias e serviços seria necessário reutilizar, e assim manter uma coerência entre as aplicações existentes e o atual projeto.

Numa primeira fase, foi feito um levantamento dos requisitos e dos diferentes casos de uso da aplicação. Cada caso de uso foi devidamente estudado e explicado, de modo a tentar considerar todas as possibilidades de ações disponíveis. A definição destes permitiu delinear a arquitetura funcional da aplicação e desenvolver a modelação dos diferentes modelos de dados. Esta etapa demonstrou ser bastante desafiadora, uma vez que, com o progresso do desenvolvimento da aplicação, foi necessário efetuar pequenos ajustes em algumas tabelas, para assim permitir a persistência de dados que inicialmente não se tinham tido em consideração. Antes de se partir para a escrita de código, foram definidos os diagramas de sequência, que permitem perceber a lógica por detrás dos diferentes casos de uso, e assim facilitar posteriormente o processo de desenvolvimento.

Relativamente ao desenvolvimento da aplicação, houve alguns obstáculos que foi necessário ultrapassar. Uma vez que foi necessário reutilizar alguns dos serviços previamente desenvolvidos pela *Altice*, foi necessário configurar o ambiente de desenvolvimento inicial, para permitir ser possível comunicar com os ditos serviços. Adicionalmente, foi necessário colaborar com alguns elementos da equipa de *frontend* da *Altice*, para que a interface da aplicação seguisse os mesmos padrões utilizados na empresa. Um outro obstáculo residiu na compreensão da biblioteca *Web Bluetooth*, utilizada para se proceder à conexão com os aparelhos utilizados pelos pacientes. Uma vez que se trata de uma tecnologia com pouca documentação, foi necessário dispendir algum tempo para perceber esta, e averiguar se se enquadrava com os objetivos pretendidos.

Uma vez concluída a parte teórica, foi possível avançar para a parte mais desafiante, a parte prática. Um dos principais focos foi garantir que o produto final respeitava os requisitos funcionais e não-funcionais definidos. Tal foi garantido pelas sucessivas validações semanais que eram realizadas.

Em retrospectiva, conclui-se que a aplicação desenvolvida cumpre o seu intuito, apoiada também por uma interface gráfica apelativa e que cumpre os princípios de usabilidade pretendidos. Adicionalmente, e devido à forma abstrata como a conexão *Bluetooth* foi desenvolvida, é possível adicionar compatibilidade com novos aparelhos *Bluetooth* de forma bastante fácil, o que se considera ser uma das vantagens do presente projeto.

Tendo todos estes fatores em conta, é possível concluir que o desenvolvimento da presente dissertação foi bastante positivo, e que poderá ter um impacto significativo na área da *eHealth*, quer para pacientes, quer para as entidades médicas.

5.2 Perspetiva de trabalho futuro

Tal como foi referido, apesar de todas as funcionalidades pretendidas terem sido implementadas, é sempre possível melhorar ainda mais cada projeto. No caso da aplicação atual, existem certos aspetos que ajudariam a contribuir para um projeto mais sólido/robusto.

Primeiramente, seria possível tornar o sistema mais responsivo para certas dimensões de aparelhos, nomeadamente telemóveis. Uma outra funcionalidade deveria ser permitir que se pudesse enviar o *PDF* do relatório por *email*, para um certo médico, sem ser necessário sair da aplicação. Em termos de segurança, podem ser implementadas medidas extra, tais como a utilização do *JWT* do cliente na comunicação entre serviços, permitindo limitar o acesso a outros serviços a partir de um serviço cuja segurança ficou comprometida ou então implementar a comunicação entre serviços com *SSL* e certificados privados.

Por fim, poderia ser permitido um paciente alterar alguns tipos de informação, como doenças, alergias, entre outras.

Bibliografia

- [1] SNS 24. App Telemonit SNS 24. URL <https://www.sns24.gov.pt/guia/app-telemonit-sns-24/>. [Online; acedido em 15/12/2022].
- [2] American Heart Association. Types of blood pressure medications. URL <https://www.heart.org/en/health-topics/high-blood-pressure/changes-you-can-make-to-manage-high-blood-pressure/types-of-blood-pressure-medications#ACE>. [Online; acedido em 11/12/2022].
- [3] Portal Nacional de Saúde. Definição doença crónica, . URL <http://pns.dgs.pt/files/2010/09/ddc.pdf>. [Online; acedido em 11/12/2022].
- [4] Serviço Nacional de Saúde. Hipertensão arterial, . URL <https://www.sns24.gov.pt/tema/doencas-do-coracao/hipertensao-arterial/>. [Online; acedido em 11/12/2022].
- [5] Biblioteca Virtual em Saúde. Dia nacional da saúde. URL <https://bvsms.saude.gov.br/05-8-dia-nacional-da-saude/>. [Online; acedido em 10/12/2022].
- [6] Expresso. Como está a saúde dos portugueses e das portuguesas? metade respondeu “bom” ou “muito bom” ao ine. URL <https://expresso.pt/sociedade/2022-02-25-como-esta-a-saude-dos-portugueses-e-das-portuguesas--metade-respondeu-bom-ou-muito-bom-ao-ine>. [Online; acedido em 11/12/2022].
- [7] Mostafa Haghi Kashani, Mona Madanipour, Mohammad Nikravan, Parvaneh Asghari, and Ebrahim Mahdipour. A systematic review of iot in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications*, 192:103164, 2021. ISSN 1084-8045. doi: <https://doi.org/10.1016/j.jnca.2021.103164>. URL <https://www.sciencedirect.com/science/article/pii/S1084804521001764>.
- [8] Martha N. Hill, Nancy Houston Miller, and Sabina DeGeest. Adherence and persistence with taking medication to control high blood pressure. *Journal of the American Society of Hypertension*, 5

- (1):56–63, 2011. ISSN 1933-1711. doi: <https://doi.org/10.1016/j.jash.2011.01.001>. URL <https://www.sciencedirect.com/science/article/pii/S1933171111000027>.
- [9] HomeDoctor. URL <https://homedoctor.es/en/>. [Online; acedido em 16/12/2022].
- [10] HopeCare. Hcalert, . URL <https://hope-care.pt/pt/hc-alert/>. [Online; acedido em 16/12/2022].
- [11] HopeCare, . URL <https://hope-care.pt/pt/>. [Online; acedido em 15/12/2022].
- [12] Altice Labs. Site altice labs, . URL <https://www.alticelabs.com/our-dna/>. [Online; acedido em 10/12/2022].
- [13] Altice Labs. Medigraf, . URL <https://www.alticelabs.com/products/ehealth-telemedicine/>. [Online; acedido em 10/12/2022].
- [14] Altice Labs. Smartal, . URL <https://www.alticelabs.com/products/ehealth-smart-assisted-living/>. [Online; acedido em 10/12/2022].
- [15] Lucy Lamboley. Examples of remote patient monitoring: 9 top patient applications. URL <https://blog.prevounce.com/examples-of-remote-patient-monitoring-9-top-patient-applications>. [Online; acedido em 10/12/2022].
- [16] Lakmini P Malasinghe, Naeem Ramzan, and Keshav Dahal. Remote patient monitoring: a comprehensive study. *Journal of Ambient Intelligence and Humanized Computing*, 10:76, 2019. ISSN 1868-5145. URL <https://doi.org/10.1007/s12652-017-0598-x>.
- [17] Sic Notícias. Portugal é o país da UE que está a envelhecer mais rápido. 21 de agosto de 2022. URL <https://sicnoticias.pt/pais/2022-08-21-Portugal-e-o-pais-da-UE-que-esta-a-envelhecer-mais-rapido-7de5ed72>. [Online; acedido em 11/12/2022].
- [18] Observador. Inspeção-geral da saúde investiga cobrança de sensores para monitorizar glicemia. 12 de dezembro de 2022. URL <https://observador.pt/2022/11/18/inspecao-geral-da-saude-investiga-cobranca-de-sensores-para-monitorizar-glicemia/>. [Online; acedido em 12/12/2022].

- [19] Hans Oh, Carlos Rizo, Murray Enkin, and Alejandro Jadad. What is ehealth (3): A systematic review of published definitions. *J Med Internet Res*, 7:e1, 2005. ISSN 1438-8871. doi: 10.2196/jmir.7.1.e1. URL <http://www.ncbi.nlm.nih.gov/pubmed/15829471>.
- [20] Jornal Público. Oms: quase 60% dos adultos na europa vivem com excesso de peso ou obesidade. URL <https://www.publico.pt/2022/05/03/ciencia/noticia/quase-60-adultos-regiao-europeia-oms-vivem-excesso-peso-obesidade-2004696>. [Online; acedido em 11/12/2022].
- [21] Nações Unidas. Envelhecimento. URL <https://unric.org/pt/envelhecimento/>. [Online; acedido em 11/12/2022].
- [22] Óbidos Diário. Solução tecnológica da hope care reduz número de internamentos em dois hospitais do país. URL <https://obidosdiario.com/2018/11/20/solucao-tecnologica-da-hope-care-reduz-numero-de-internamentos-em-dois-hospitais-do-pais/>. [Online; acedido em 12/12/2022].

Anexo A

Requisitos Funcionais

Requisito: RF01

Descrição: O utilizador deve conseguir iniciar a sessão na aplicação.

Fundamentação: Para um utilizador aceder ao sistema, terá de iniciar sessão na aplicação.

Ator: Utilizador

Origem: Reunião

Prioridade: *Must*

Requisito: RF02

Descrição: O utilizador deve conseguir ver perfil.

Fundamentação: Qualquer utilizador deverá ter acesso ao sue perfil e visualizar as suas informações.

Ator: Utilizador

Origem: Reunião

Prioridade: *Must*

Requisito: RF03

Descrição: O paciente deve conseguir realizar medições clínicas.

Fundamentação: Um paciente pode registar medições de forma manual ou através de dispositivos Bluetooth.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF04

Descrição: O paciente deve conseguir visualizar as suas medições clínicas.

Fundamentação: Todas as medições efetuadas por um paciente deverão estar disponíveis na aplicação.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF05

Descrição: O paciente deve conseguir emparelhar dispositivos Bluetooth com a aplicação.

Fundamentação: Caso o paciente pretenda, poderá conectar os seus dispositivos com o sistema.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF06

Descrição: O paciente deve conseguir inserir medições manualmente.

Fundamentação: O sistema deve permitir que valores de medições sejam inseridos manualmente.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF07

Descrição: O paciente deve conseguir comentar medições realizadas.

Fundamentação: Na eventualidade de querer deixar alguma nota acerca da medição efetuada, o paciente poderá deixar um comentário.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF08

Descrição: O paciente deve conseguir criar PDF com histórico de medições pretendidas.

Fundamentação: O paciente poderá selecionar as medidas pretendidas e obter um PDF com as medições associadas a essas medidas, num determinado intervalo de tempo.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF09

Descrição: O paciente deve conseguir editar as suas *thresholds*.

Fundamentação: Cada *threshold* pode ser modificada para se ajustar ao estado de saúde de um paciente.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF10

Descrição: O paciente deve conseguir adicionar médicos para partilha de informação.

Fundamentação: As medições do paciente podem ser partilhadas com os médicos que este pretender.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF11

Descrição: O paciente deve conseguir remover médicos de partilha de informação.

Fundamentação: As medições do paciente podem ser partilhadas com os médicos que este pretender.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF12

Descrição: O paciente deve conseguir responder a questionário.

Fundamentação: Para fornecer mais *feedback* sobre o seu estado de saúde, o paciente deve conseguir preencher os questionários publicados pelos médicos.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF13

Descrição: O paciente deve conseguir visualizar as suas respostas a questionário.

Fundamentação: O paciente deve conseguir ver as suas respostas seleccionadas, quer na sessão atual, quer em sessões anteriores.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF14

Descrição: O paciente deve conseguir retomar a resolução de questionário.

Fundamentação: Na eventualidade de ter deixado a resolução de um questionário a meio, o paciente poderá retomar esta a qualquer momento.

Ator: Paciente

Origem: Reunião

Prioridade: *Must*

Requisito: RF15

Descrição: O médico deve conseguir visualizar os seus pacientes.

Fundamentação: Um médico deve saber quais os seus pacientes.

Ator: Médico

Origem: Reunião

Prioridade: *Must*

Requisito: RF16

Descrição: O médico deve conseguir visualizar as últimas medições de um paciente.

Fundamentação: As últimas medições de um paciente deverão ser facilmente acedidas por parte do médico.

Ator: Médico

Origem: Reunião

Prioridade: *Must*

Requisito: RF17

Descrição: O médico deve conseguir visualizar histórico de medições de um paciente.

Fundamentação: Caso pretenda, o médico poderá consultar todas as medições efetuadas pelos seus pacientes, num determinado período.

Ator: Médico

Origem: Reunião

Prioridade: *Must*

Requisito: RF18

Descrição: O médico deve conseguir editar *thresholds* de um paciente.

Fundamentação: Um médico poderá editar as *thresholds* de um dos seus pacientes.

Ator: Médico

Origem: Reunião

Prioridade: *Must*

Requisito: RF19

Descrição: O médico deve conseguir realizar o *upload* de um questionário.

Fundamentação: Um médico poderá adicionar questionários ao sistema, que posteriormente poderão ser preenchidos pelos seus pacientes.

Ator: Médico

Origem: Reunião

Prioridade: *Must*

Requisito: RF20

Descrição: O médico deve conseguir eliminar um questionário.

Fundamentação: Sempre que pretenda, um médico pode eliminar um questionário do sistema

Ator: Médico

Origem: Reunião

Prioridade: *Must*

Anexo B

Requisitos Não Funcionais

Requisito: RNF01

Descrição: As páginas devem ser totalmente carregadas em menos de 1,5 segundos.

Fundamentação: A aplicação deve carregar as páginas de forma rápida.

Origem: Reunião

Critério de ajuste: Em 90% dos caso, a aplicação deve carregar qualquer página em menos de 1,5 segundos.

Prioridade: *Should*

Requisito: RNF02

Descrição: A aplicação deve ter as funcionalidades acessíveis em poucos cliques.

Fundamentação: A aplicação deve carregar as páginas de forma rápida.

Origem: Introspeção

Critério de ajuste: O utilizador deve conseguir aceder às diversas funcionalidades do sistema em menos de 5 cliques.

Prioridade: *Should*

Requisito: RNF03

Descrição: O sistema deve impedir o progresso em caso de dados inválidos.

Fundamentação: Garantir que o utilizador introduz dados válidos.

Origem: Introspeção

Critério de ajuste: Nenhum dado errado poderá ser submetido pelo utilizador.

Prioridade: *Should*

Requisito: RNF04

Descrição: O sistema deve notificar quando uma ação é concluída com sucesso.

Fundamentação: Fornecer *feedback* sempre que uma ação é executada.

Origem: Introspeção

Critério de ajuste: Notificação aparece no ecrã sempre que uma ação é concluída.

Prioridade: *Should*

Requisito: RNF05

Descrição: O sistema deve ser de fácil utilização para utilizadores sem experiência.

Fundamentação: Para garantir uma boa experiência de utilização.

Origem: Introspeção

Critério de ajuste: 90% dos utilizadores sem experiência devem ser capazes de aceder às funcionalidades principais em menos de 30 segundos.

Prioridade: *Should*

Requisito: RNF06

Descrição: O sistema deve suportar um vasto leque de dispositivos *Bluetooth*.

Fundamentação: Garantir uma maior qualidade do sistema.

Origem: Introspeção

Critério de ajuste: A aplicação deve ser compatível com pelo menos 10 dispositivos *Bluetooth*.

Prioridade: *Should*

Requisito: RNF07

Descrição: O sistema deve encriptar os dados dos utilizadores.

Fundamentação: Maximizar a segurança no sistema.

Origem: Introspeção

Critério de ajuste: As *passwords* devem ser sempre encriptadas.

Prioridade: *Should*

Anexo C

Use Cases

C.1 Iniciar sessão

De modo a um utilizador ter acesso ao sistema, é necessário que as suas credencias estejam previamente armazenadas na base de dados de aplicação. Caso tal não se suceda, e o utilizador ainda não possua conta, então este deverá registar-se.

Use Case	Iniciar Sessão
Descrição	O utilizador inicia sessão no sistema.
Ator	Médico, Paciente
Pré-Condição	<ol style="list-style-type: none">1. O utilizador possui conta no sistema.2. O utilizador encontra-se na página de iniciar sessão.
Pós-Condição	O utilizador é reencaminhado para a página principal do sistema.
Fluxo Normal	<ol style="list-style-type: none">1. O utilizador insere as suas credenciais.2. O sistema valida as credenciais.3. O utilizador é reencaminhado para a página principal do sistema.
Fluxo Exceção [2.1]	<ol style="list-style-type: none">2.1. O sistema não consegue validar as credenciais.2.2. O sistema apresenta uma mensagem de erro.

Figura 55: Use Case - Iniciar Sessão

C.2 Editar thresholds

Apesar de haver um conjunto inicial de *thresholds* associado a cada tipo de dados, poderá ser necessário ajustar alguns deles individualmente, para que melhor se adequem ao estado de saúde de um determinado paciente. Tal deve-se ao facto de, em certos cenários, uma medição poder ser normal para certos pacientes, e no entanto ser de risco para um paciente com algum problema de saúde mais grave.

Use Case	Editar <i>thresholds</i>
Descrição	O paciente edita as suas <i>thresholds</i> para um determinado tipo de dados.
Ator	Paciente
Pré-Condição	O paciente está autenticado na aplicação.
Pós-Condição	As novas <i>thresholds</i> são apresentadas no ecrã.
Fluxo Normal	<ol style="list-style-type: none"> 1. O paciente seleciona a opção de editar <i>thresholds</i>. 2. O sistema obtém todas as <i>thresholds</i> do paciente. 3. Os <i>thresholds</i> de cada tipo são apresentados. 4. O paciente altera um determinado <i>threshold</i>. 5. O paciente confirma as alterações. 6. O sistema regista as novas <i>thresholds</i> do paciente. 7. As novas <i>thresholds</i> são apresentadas no ecrã.

Figura 56: Use Case - Editar *thresholds*

C.3 Consultar medições

Todas as medições efetuadas por um paciente deverão poder ser visualizadas por este mesmo. As medições que merecem destaque são as últimas a ser realizadas, e portanto, de modo a facilitar a leitura das últimas medições, a interface principal deverá apresentar, numa primeira instância, essas mesmas. Adicionalmente, o utilizador poderá ter interesse em consultar todas as medições referentes a um determinado grupo, num determinado período, e por isso deverá ser possível ao paciente selecionar um desses grupos e visualizar então o histórico de medições associado a esse mesmo. O histórico poderá ser visualizado quer sob a forma de gráfico, quer sob a forma de tabela.

Use Case	Consultar medições
Descrição	O paciente visualiza o histórico de medições de um determinado tipo de medição.
Ator	Paciente
Pré-Condição	<ol style="list-style-type: none"> 1. O paciente está autenticado na aplicação. 2. O paciente já realizou medições previamente. 3. O paciente encontra-se na <i>tab</i> das suas medições
Pós-Condição	As medições são apresentadas no ecrã.
Fluxo Normal	<ol style="list-style-type: none"> 1. O sistema obtém as últimas medições do paciente 2. O paciente visualiza as suas medições.
Fluxo Alternativo 1 [1.1]	<ol style="list-style-type: none"> 2.1. O paciente seleciona a medição pretendida. 2.2. O paciente seleciona o período de visualização pretendido. 2.3. O sistema obtém as medições efetuadas durante esse período. 2.4. O paciente escolhe a opção de leitura por gráfico. 2.5. As medições são apresentadas no ecrã sob a forma de gráfico.
Fluxo Alternativo 2 [1.3.1]	<ol style="list-style-type: none"> 2.4.1. O paciente escolhe a opção de leitura por tabela. 2.4.2. As medições são apresentadas no ecrã sob a forma de tabela.

Figura 57: Use Case - Consultar medições

C.4 Gerar PDF do histórico

Poderá haver cenários em que um determinado médico pretendido pelo paciente não se encontre na plataforma. De tal modo, e para ser possível apresentar as medições efetuadas ao médico pretendido, o sistema deverá permitir que seja gerado um documento PDF com as informações selecionadas pelo paciente. Para tal, este terá de selecionar o período de tempo pretendido, e quais os grupos de dados que pretende que sejam visíveis. Adicionalmente, as últimas medições de cada grupo também serão apresentadas no documento.

Use Case	Gerar PDF do histórico
Descrição	O paciente gera um PDF com as suas últimas medições realizadas e com o histórico dos grupos de medições efetuados num determinado período.
Ator	Paciente
Pré-Condição	O paciente está autenticado na aplicação.
Pós-Condição	O PDF gerado é apresentado no ecrã.
Fluxo Normal	<ol style="list-style-type: none">1. O paciente seleciona a opção de gerar PDF.2. É apresentado um modal com os grupos que pretende visualizar e o período pretendido.3. O paciente seleciona os grupos e o período.4. O paciente gera o PDF.5. O sistema obtém as medições dos grupos selecionados.6. O PDF gerado é apresentado no ecrã.

Figura 58: Use Case - Gerar PDF do histórico

C.5 Responder a questionário

Os questionários disponibilizados pelos médicos estarão acessíveis para todos os pacientes. Os questionários são utilizados para que os médicos possam ter uma melhor noção do estado de saúde do paciente, e consigam proceder em caso de necessidade.

Sempre que um utilizador pretende responder a um questionário, será criada uma nova sessão associada a esse mesmo questionário, que simboliza uma nova tentativa. Durante a resolução do questionário, caso o paciente saia a meio, a tentativa ficará registada como incompleta, e o paciente poderá resumir a resolução assim que pretender. Quando todas as respostas ao questionário forem submetidas, a sessão fica declarada como encerrada, podendo depois o utilizador visualizar quais as respostas que selecionou. Adicionalmente, nos casos em que um paciente inicia a resolução de um questionário ou submete as suas respostas finais, será enviada uma notificação para os seus médicos, que assim ficam a par das ações realizadas pelo paciente.

Use Case	Responder a questionário
Descrição	O paciente responde a um questionário.
Ator	paciente
Pré-Condição	<ol style="list-style-type: none"> 1. O paciente está autenticado na aplicação. 2. O paciente encontra-se na <i>tab</i> dos seus questionários. 3. Existem questionários disponíveis.
Pós-Condição	A sessão atual fica registada no sistema
Fluxo Normal	<ol style="list-style-type: none"> 1. O paciente escolhe o questionário pretendido. 2. O paciente escolhe iniciar uma nova sessão de resposta ao questionário. 3. O sistema obtém as respostas do questionário. 4. As questões do questionário são apresentadas pela devida ordem. 5. O paciente responde a todas as questões e submete as suas respostas. 6. O sistema guarda as respostas dadas. 7. A sessão atual fica registada como submetida no sistema.
Fluxo Alternativo 1 [4.1]	<ol style="list-style-type: none"> 6.1. O paciente não responde a todas as questões. 6.2. A sessão atual fica registada como incompleta, e o utente pode continuar a responder a este quando pretender.
Fluxo Alternativo 2 [4.1.1]	<ol style="list-style-type: none"> 4.1.1. O paciente responde a todas as questões e submete as suas respostas. 4.1.2. A sessão atual fica registada como submetida no sistema.

Figura 59: Use Case - Responder a questionário

C.6 Consultar questionário

Tal como foi dito anteriormente, o paciente poderá visualizar as respostas selecionadas numa determinada submissão. Esta funcionalidade permite que o paciente possa analisar a sua submissão sempre que pretenda, e consiga também perceber se as suas respostas têm ou não divergido muito de sessão para sessão.

Use Case	Consultar questionário
Descrição	O paciente visualiza as suas respostas a um questionário.
Ator	Paciente
Pré-Condição	<ol style="list-style-type: none"> 1. O paciente está autenticado na aplicação. 2. Existem questionários disponíveis. 3. O paciente encontra-se na <i>tab</i> dos seus questionários. 4. O paciente respondeu ao questionário pretendido.
Pós-Condição	As respostas ao questionário são apresentadas na aplicação.
Fluxo Normal	<ol style="list-style-type: none"> 1. O sistema obtém todos os questionários disponíveis. 2. O paciente escolhe o questionário pretendido 3. O paciente seleciona a sessão que pretende consultar. 4. O sistema obtém as respostas dadas nessa mesma sessão. 5. As respostas escolhidas são apresentadas na aplicação.

Figura 60: Use Case - Consultar questionário

C.7 Definir médicos de partilha

Apesar de um paciente ter acesso aos seus médicos na plataforma, poderá querer apenas partilhar informação com algum deles. De tal modo, é importante haver uma secção este possa seleccionar, por especialidade, quais os médicos que podem ter acesso às suas informações, desde medições efetuadas, até às respostas a questionários.

Use Case	Definir médicos de partilha
Descrição	O paciente seleciona os médicos que podem ter acesso às suas medições.
Ator	Paciente
Pré-Condição	O paciente está autenticado na aplicação.
Pós-Condição	Os médicos selecionados aparecem visualmente destacados.
Fluxo Normal	<ol style="list-style-type: none"> 1. O paciente seleciona a opção de consultar os seus médicos 2. O sistema obtém todas as especialidades. 3. O paciente seleciona a especialidade dos médicos que pretende consultar. 4. O sistema obtém os médicos dessa mesma especialidade. 5. O paciente seleciona um novo médico. 6. O paciente confirma a sua seleção. 7. O sistema adiciona os privilégios de partilha ao médico selecionado. 8. O novo médico aparece visualmente destacado. 9. Os médicos selecionados aparecem visualmente destacados.
Fluxo Alternativo [3.1]	<ol style="list-style-type: none"> 7.1. O paciente seleciona o médico que pretende remover. 7.2. O paciente confirma a sua seleção. 7.3. O sistema remove os privilégios de partilha ao médico selecionado. 7.4. O médico escolhido deixa de aparecer visualmente destacado. 7.5. Voltar a [9]

Figura 61: Use Case - Definir médicos de partilha

C.8 Editar thresholds de paciente

Um médico que tenha acesso às informações de um paciente poderá também editar as *thresholds* desse mesmo paciente. Tal funcionalidade permite, para utilizadores mais incapacitados, que seja o próprio médico a agir consoante as alterações do estado de saúde deste.

Use Case	Editar <i>thresholds</i> de paciente
Descrição	O médico edita as <i>thresholds</i> de um dado paciente.
Ator	Médico
Pré-Condição	<ol style="list-style-type: none"> 1. O médico está autenticado na aplicação. 2. O médico tem privilégios de partilha com o paciente. 3. O médico encontra-se na <i>tab</i> dos seus pacientes.
Pós-Condição	As novas <i>thresholds</i> são apresentadas no ecrã.
Fluxo Normal	<ol style="list-style-type: none"> 1. O médico seleciona o paciente pretendido. 2. O sistema obtém as <i>thresholds</i> do paciente. 3. O médico altera um determinado <i>threshold</i>. 4. O médico confirma as alterações. 5. O sistema regista as novas <i>thresholds</i> do paciente. 6. As novas <i>thresholds</i> são apresentadas no ecrã.

Figura 62: Use Case - Editar thresholds de paciente

C.9 Consultar respostas de paciente a questionário

Adicionalmente, o médico pode consultar as respostas de um determinado paciente a um questionário. Para tal, quando se encontrar na secção dos seus pacientes, terá de escolher a opção referente aos questionários e ver todas as submissões realizadas pelo paciente selecionado.

Use Case	Consultar respostas de paciente a questionário
Descrição	O médico visualiza as respostas de um paciente a um determinado questionário.
Ator	Médico
Pré-Condição	<ol style="list-style-type: none"> 1. O médico está autenticado no sistema. 2. Existem questionários no sistema. 3. O paciente já respondeu ao questionário. 4. O médico encontra-se na <i>tab</i> dos seus pacientes e seleciona a opção de medições.
Pós-Condição	As respostas ao questionário são apresentadas.
Fluxo Normal	<ol style="list-style-type: none"> 1. O médico seleciona o paciente pretendido. 2. O sistema obtém as informações do paciente. 3. O médico seleciona a opção de questionários. 4. O sistema adquire os questionários respondidos pelo paciente. 5. Os questionários respondidos são apresentados. 6. O médico seleciona a secção que pretende visualizar. 7. O sistema obtém as perguntas desse questionário. 8. As respostas ao questionário são apresentadas.

Figura 63: Use Case - Consultar questionário

C.10 Realizar upload de questionário

Sempre que um novo questionário estiver disponível para ser disponibilizado aos pacientes, um médico poderá proceder à publicação deste na plataforma. Os questionários originais veem em formato .csv, e devem seguir um conjunto de regras específicas. Tal documento é depois convertido para um objeto *json* e modificado de forma a poder ser respondido pelos pacientes.

Use Case	Realizar <i>upload</i> de questionário
Descrição	O médico pretende dar <i>upload</i> de um novo questionário para o sistema.
Ator	Médico
Pré-Condição	<ol style="list-style-type: none">1. O médico está autenticado no sistema.2. O médico encontra-se na <i>tab</i> de questionários
Pós-Condição	O novo questionário aparece disponibilizado no sistema.
Fluxo Normal	<ol style="list-style-type: none">1. O médico seleciona a opção de upload de questionário.2. O médico seleciona o ficheiro .csv no seu dispositivo.3. O médico faz <i>upload</i> desse mesmo ficheiro.4. O sistema valida que o questionário ainda não existe.5. O novo questionário aparece disponibilizado no sistema.
Fluxo Exceção [4.1]	<ol style="list-style-type: none">4.1. O sistema verifica que o questionário já existe.4.2. É apresentada uma mensagem de erro.4.3. O questionário não é adicionado ao sistema.

Figura 64: Use Case - Realizar upload de questionário

C.11 Eliminar questionário

Sempre que pretender, um médico poderá eliminar um questionário do sistema. Sempre que um questionário é eliminado, todas as sessões associadas a esse mesmo serão eliminadas, ou seja, as respostas escolhidas pelos pacientes nas diferentes sessões serão removidas do sistema, passando a ser impossíveis de aceder por parte dos pacientes.

Use Case	Eliminar questionário
Descrição	O médico pretende eliminar um questionário do sistema.
Ator	Médico
Pré-Condição	<ol style="list-style-type: none"> 1. O médico está autenticado no sistema. 2. O médico encontra-se na <i>tab</i> de questionários. 3. Existem questionários no sistema.
Pós-Condição	O questionário é eliminado do sistema.
Fluxo Normal	<ol style="list-style-type: none"> 1. O médico seleciona o questionário pretendido. 2. O médico seleciona a opção de eliminar questionário. 3. O sistema pede a confirmação da decisão. 4. O sistema elimina o questionário do sistema. 5. O questionário é eliminado do sistema

Figura 65: Use Case - Eliminar questionário

C.12 Consultar template de questionário

Todos os questionários adicionados pelos médicos poderão ser consultados por eles mesmos, podendo verificar se se encontra tudo em ordem ou se há falta de alguma informação importante.

Use Case	Consultar <i>template</i> de questionário
Descrição	O médico visualiza a estrutura de um questionário.
Ator	Médico
Pré-Condição	<ol style="list-style-type: none"> 1. O médico está autenticado no sistema. 2. O médico encontra-se na <i>tab</i> dos questionários. 3. Existem questionários no sistema.
Pós-Condição	As perguntas do questionário são apresentadas.
Fluxo Normal	<ol style="list-style-type: none"> 1. O médico seleciona o questionário pretendido. 2. O sistema adquire as perguntas desse questionário. 3. As perguntas do questionário são apresentadas.

Figura 66: Use Case - Consultar template de questionário

C.13 Ver perfil

Cada utilizador possuirá uma secção referente ao seu perfil, onde poderá consultar as suas informações pessoais. Dependendo do tipo de utilizador, isto é, se é paciente ou médico, as informações apresentadas serão diferentes.

Use Case	Ver perfil
Descrição	O utilizador visualiza o seu perfil.
Ator	Médico, Paciente
Pré-Condição	<ol style="list-style-type: none"> 1. O utilizador está autenticado no sistema. 2. O utilizador encontra-se na <i>tab</i> de seu perfil.
Pós-Condição	A informação do utilizador é apresentada no ecrã.
Fluxo Normal	<ol style="list-style-type: none"> 1. O sistema adquire a informação do utilizador. 2. A informação do utilizador é apresentada no ecrã.

Figura 67: Use Case - Ver perfil

Anexo D

Diagramas de Sequência

D.1 Editar thresholds

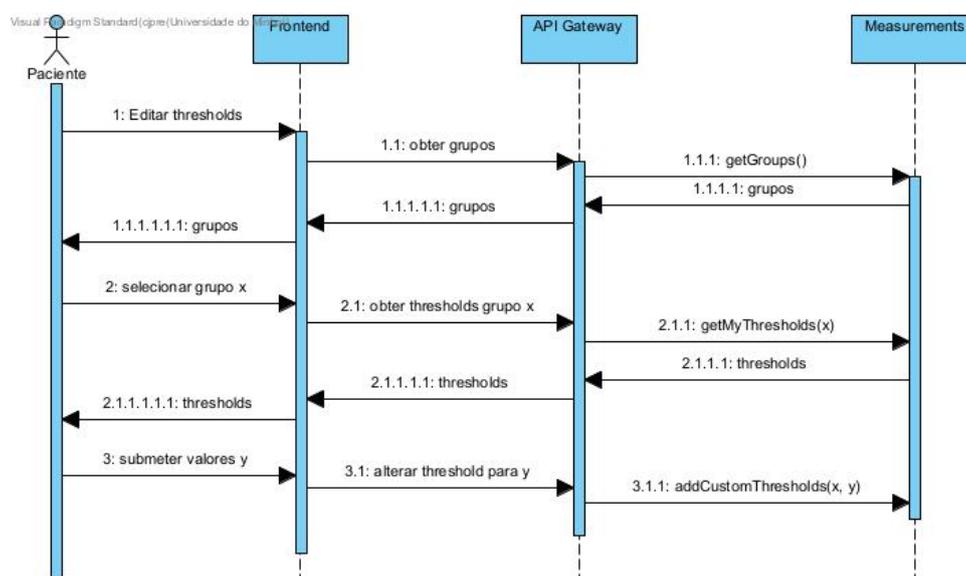


Figura 68: Diagrama Sequência - Editar thresholds

Quando um paciente quiser editar uma ou mais *thresholds*, o sistema tem de primeiro obter todos os *data_type_groups*, e só depois do paciente selecionar o grupo pretendido é que adquire as suas *thresholds*. Deste modo, evita-se carregar todas as *thresholds* de uma só vez, apenas se indo buscar aquelas selecionadas pelo utilizador. Depois de selecionado o grupo, o paciente edita algum dos valores e submete esses mesmos, a *frontend* envia à *API Gateway* o pedido com o grupo selecionado e os novos valores definidos e estes são posteriormente guardados no seu serviço.

D.2 Consultar medições

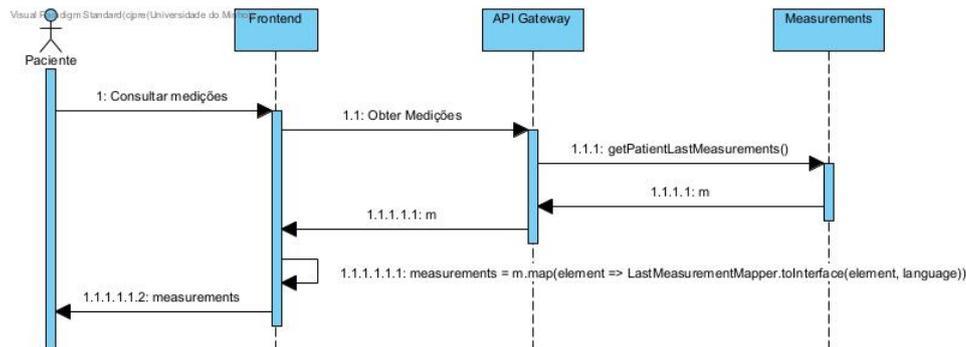


Figura 69: Diagrama Sequência - Consultar medições

Quando um paciente requisita as suas medições, para além de se ter que as ir obter ao serviço das *measurements*, também é necessário converter cada uma das medições para uma interface que possa ser utilizada para visualização no componente dos *cards*.

D.3 Gerar PDF do histórico

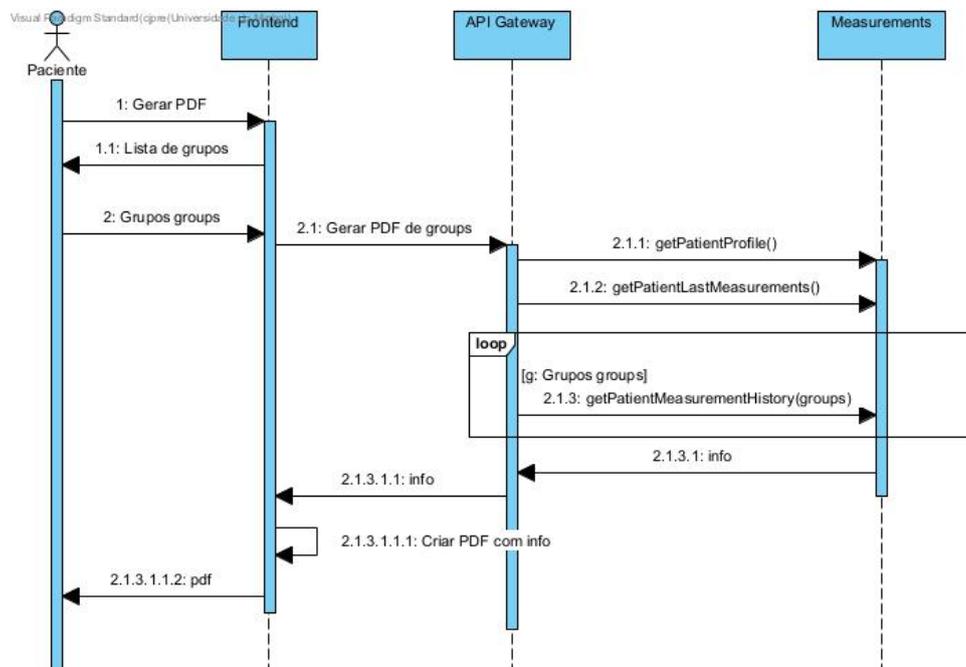


Figura 70: Diagrama Sequência - Gerar PDF do histórico

O processo de gerar o *pdf* é bastante complexo. Para além de se terem de obter todos os *data_type_groups*, também será necessário obter todas as medições compreendidas entre o intervalo de tempo selecionado.

As *lastMeasurements* precisam de ser convertidas, tal como no caso de consultar as medições, para a interface dos *cards*, enquanto que as restantes não.

D.4 Responder a questionário

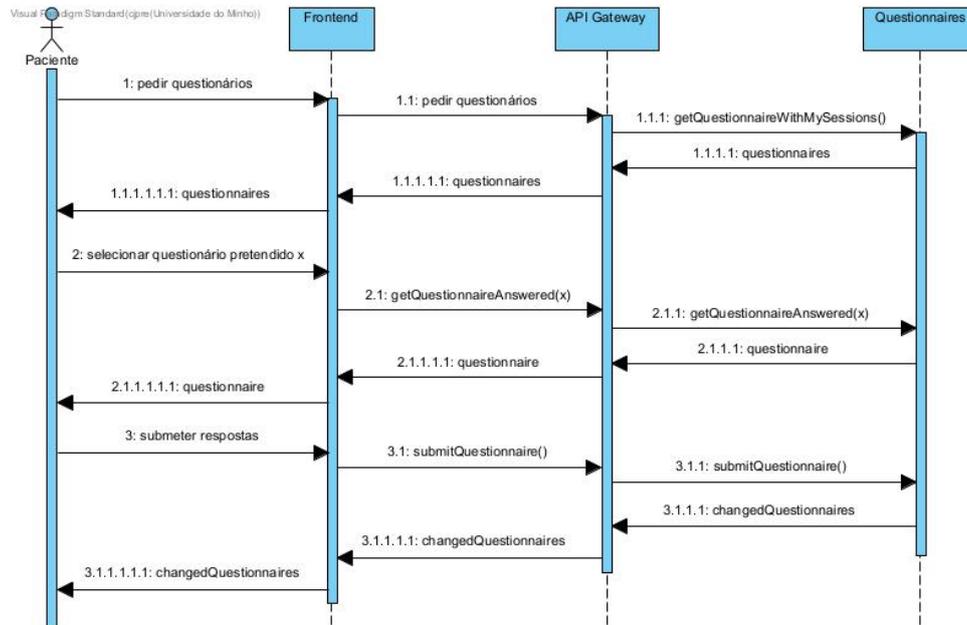


Figura 71: Diagrama Sequência - Responder a questionário

Para responder a um questionário, é primeiro necessário obter todos os questionários disponíveis, o utilizador seleciona aquele que pretende, o sistema devolve a composição deste e assim o utilizador pode não só ver todas as questões que o compõe, como submeter as suas respostas. Sempre que uma resposta é selecionada e o paciente avança para a próxima questão, a resposta é submetida, o que faz com que, caso o utilizador não tenha tempo de concluir a resolução, possa retomar do estado em que deixou o questionário da última vez.

D.5 Consultar questionário

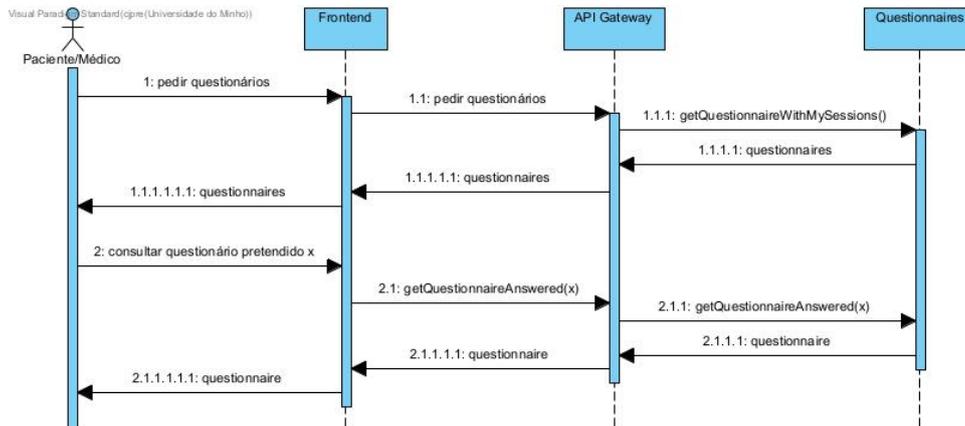


Figura 72: Diagrama Sequência - Consultar questionário

À semelhança de responder a um questionário, o sistema obtém primeiro todos os questionários disponíveis, e só depois do utilizador seleccionar aquele que quer visualizar é que se obtém esse mesmo questionário.

D.6 Definir médicos de partilha

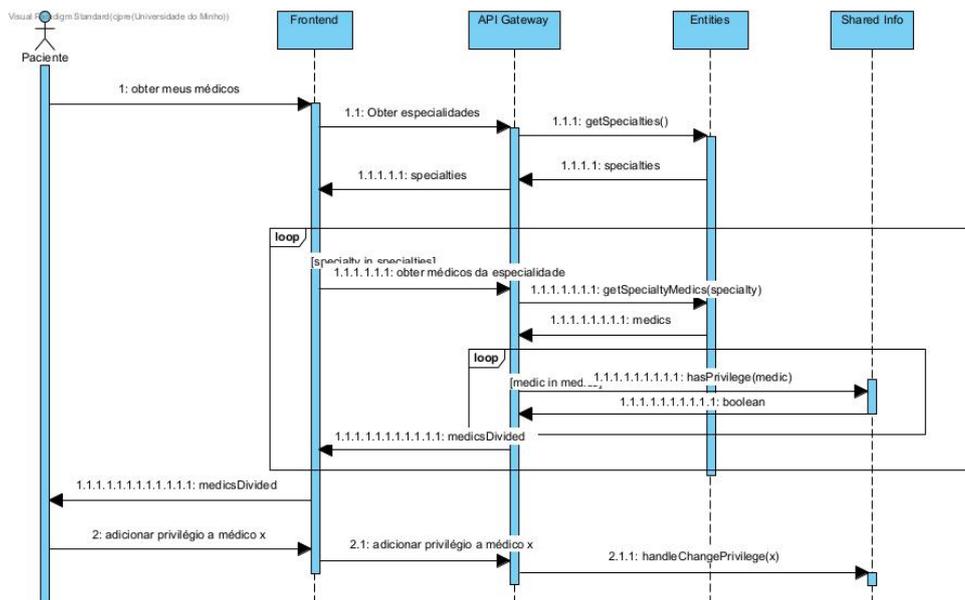


Figura 73: Diagrama Sequência - Definir médicos de partilha

Antes de se definir os médicos de partilha, o sistema tem de obter todas as especialidades, e só após o utilizador seleccionar aquela que pretende, é que se adquirem os médicos que possuem essa mesma especialidade. Para cada um desses médicos é necessário verificar se o paciente em questão forneceu privilégios de partilha ou não, para assim a *checkbox* do componente vir ou não preenchida. Quando o paciente selecciona um desses médicos, o sistema irá substituir o estado desse médico pelo seu inverso, isto é, caso o médico tivesse privilégios, esses mesmos são removidos, caso contrário são adicionados.

D.7 Editar thresholds de paciente

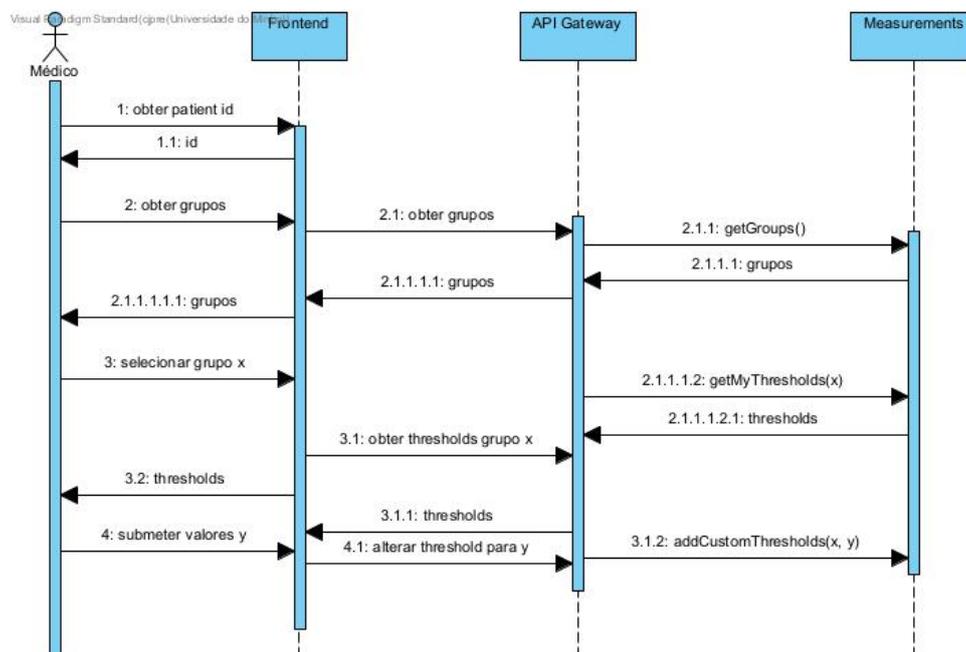


Figura 74: Diagrama Sequência - Editar thresholds de paciente

Idêntico ao modo de obter as *thresholds* por parte do paciente, só que neste cenário o médico fornece o identificador do paciente que pretende, e o sistema devolve as *thresholds* deste.

D.8 Consultar respostas de paciente a questionário

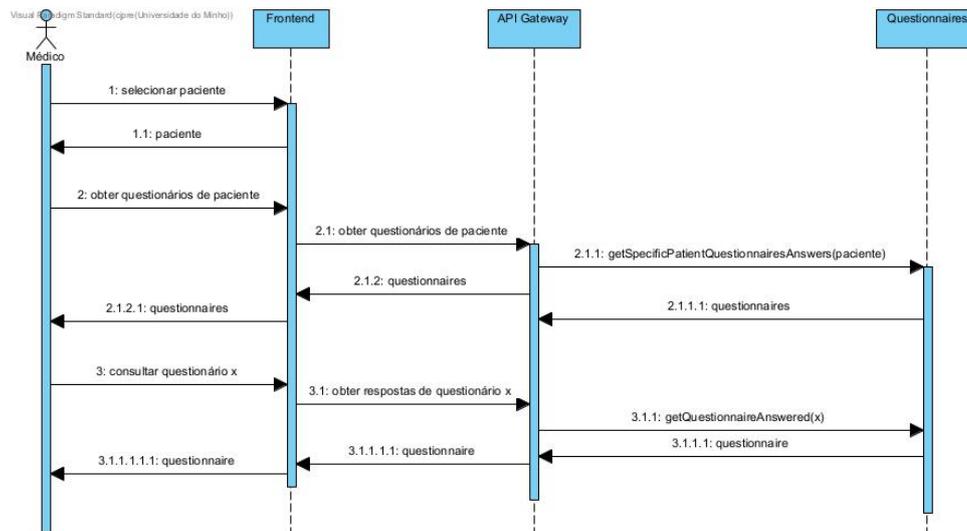


Figura 75: Diagrama Sequência - Consultar respostas de paciente a questionário

Numa primeira fase, é necessário selecionar o paciente pretendido, depois obter todos os questionários a que este tenha respondido, e só depois de selecionar qual aquele que pretende visualizar, é que o sistema adquire as respostas do paciente pretendido ao questionário selecionado.

D.9 Realizar upload de questionário

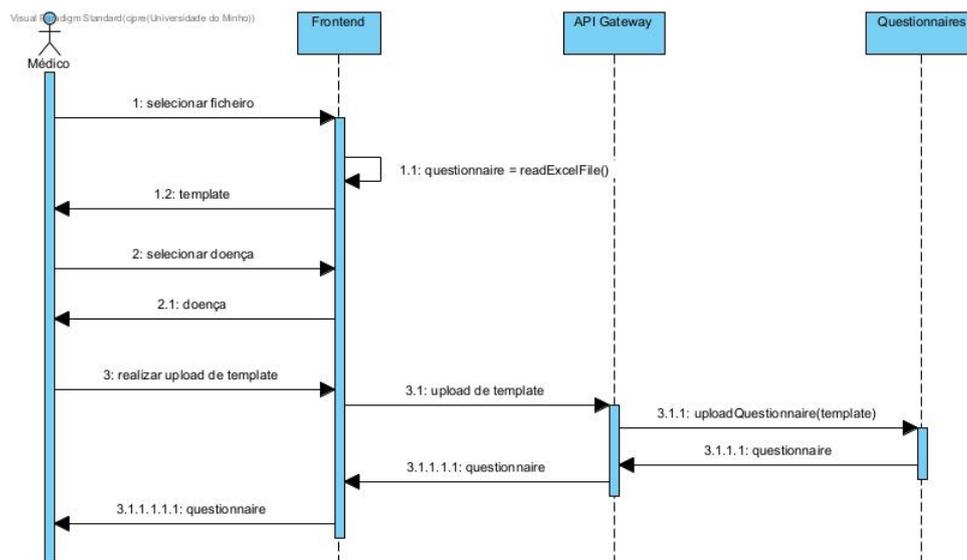


Figura 76: Diagrama Sequência - Realizar upload de questionário

Para se proceder ao *upload* de um questionário, é primeiro necessário seleccionar o ficheiro *excel* com a estrutura do questionário, seleccionar a doença referente a esse mesmo questionário, e só depois submeter para o *API Gateway*. Quando concluído, o sistema devolve o questionário adicionado, para assim este aparecer na lista dos disponíveis.

D.10 Consultar template de questionário

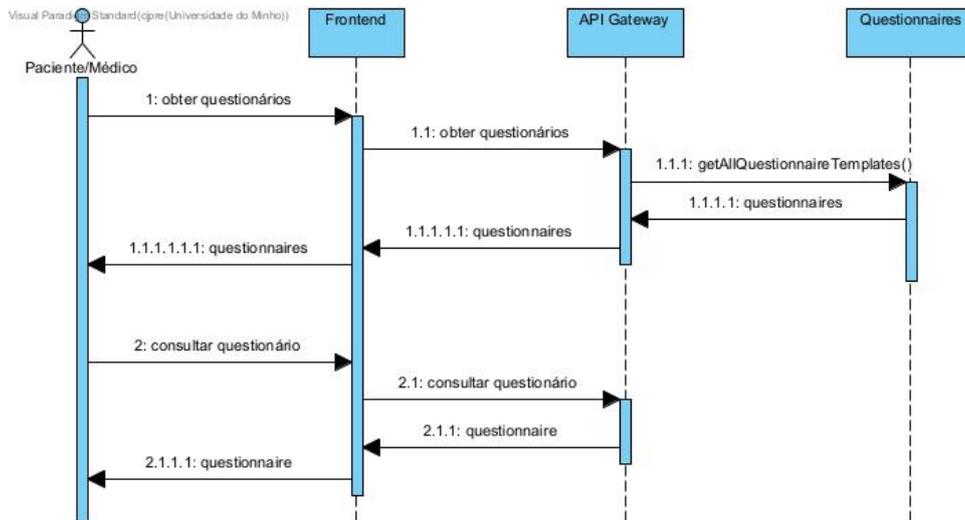


Figura 77: Diagrama Sequência - Consultar template de questionário

Para se consultar um questionário, apenas se tem de seleccionar o questionário pretendido, que o sistema devolve a sua composição.

D.11 Ver perfil

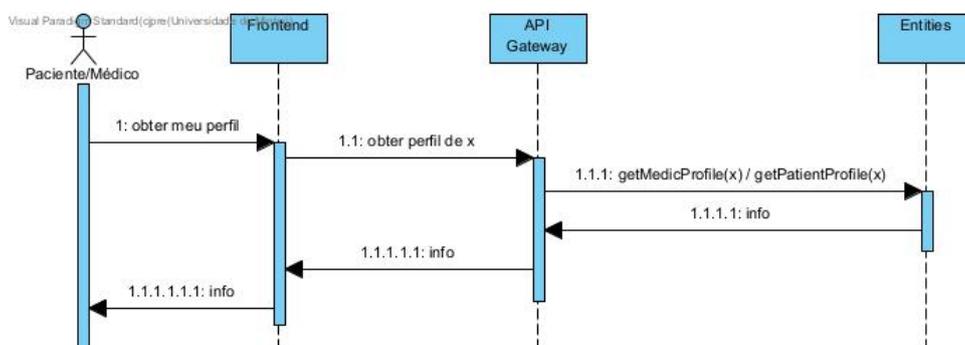


Figura 78: Diagrama Sequência - Ver perfil

Quer para o paciente, quer para o médico, o processo de obtenção do perfil é exatamente o mesmo, sendo necessário enviar um pedido ao serviço *entities* que devolve a informação do utilizador.

