

Universidade de Aveiro  
Projeto em Informática  
2019/2020

## **MyLife - Sistema Pessoal de Monitorização de Estilo de Vida**

João Marques	89234
João Vasconcelos	88808
Tiago Mendes	88886
Tomás Costa	89016
Vasco Ramos	88931

Orientador: Carlos Costa  
(Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da  
Universidade de Aveiro)



universidade de aveiro  
theoria poesis praxis



*“Os humanos são, essencialmente, o que comem, o que bem, o exercício que praticam e o ar que respiram.” - Prof. Dr. Carlos Costa*



## **Palavras-chave**

- Reconhecimento de Comida
- *Fitness Tracker*
- Análise da Ingestão de Macronutrientes
- *Machine Learning*
- *Computer Vision*
- Monitorização de Sinais Vitais
- Integração com Fitbit



## **Resumo**

O MyLife é um sistema pessoal de monitorização de estilo de vida que inclui o registo e análise de hábitos de saúde e alimentares.

O sistema contempla uma interface Web simples e intuitiva, e mecanismos ágeis de registo de atividade física e reconhecimento de alimentos com recurso a código de barras e reconhecimento automático que utiliza imagens capturadas por uma câmera e métodos de aprendizagem automática. A plataforma proposta faz uso de dispositivos de utilização generalizada como, por exemplo, pulseiras de monitorização de atividade física e telemóveis.

Em termos de engenharia de *software*, foi adoptada uma arquitetura modular que promove a extensibilidade da solução, permitindo uma fácil integração de novos dispositivos de aquisição de dados e a implementação de novas funcionalidades aplicacionais. Na prova de conceito desenvolvida, foi utilizado o dispositivo *Fitbit* para aquisição de dados relativos à atividade física e reconhecimento automático de comida com recurso a uma *Convolutional Neural Networks* (CNN).

A solução foi testada com pessoas representativas de distintos estilos de vida. Os resultados obtidos superaram as expectativas iniciais ao ponto de acreditarmos que o projeto MyLife tem um grande potencial de comercialização.



## Agradecimentos

Ao longo do desenvolvimento deste projeto, foram várias as pessoas que se disponibilizaram para nos ajudar a melhorar a qualidade do mesmo. Neste sentido, gostaríamos de agradecer:

- Ao Prof. Dr. Carlos Costa, por toda a ajuda e supervisão prestadas, em diversas vertentes, ao longo de todo o projeto.
- Ao Prof. Dr. António Sousa, ao Prof. Dr. José Moreira e ao Prof. Dr. Rui Aguiar, coordenadores da unidade curricular de Projeto em Informática.
- À Prof. Dr<sup>a</sup>. Pétia Georgieva e ao investigador João Paulo Ferreira, pelos seus esclarecimentos relacionados com o módulo de *Machine Learning*.
- Aos investigadores Micael Pedrosa e Rui Lebre, pela ajuda prestada na configuração da máquina disponibilizada no IEETA.
- E, por fim, a todos os familiares e amigos, de todos os elementos do grupo, que nos ajudaram a testar e a recolher feedback sobre o sistema.



# Conteúdo

<b>Palavras-chave</b>	<b>v</b>
<b>Resumo</b>	<b>vii</b>
<b>Agradecimentos</b>	<b>ix</b>
<b>Abreviaturas</b>	<b>xv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contexto . . . . .	1
1.2 Motivação . . . . .	1
1.3 Objetivos . . . . .	2
1.4 Estrutura do Documento . . . . .	2
<b>2 Estado da Arte e Trabalho Relacionado</b>	<b>3</b>
2.1 Monitorização de Hábitos de Saúde e Exercício Físico . . . . .	3
2.2 Reconhecimento e Classificação de Comida . . . . .	4
2.3 Reconhecimento de Exercícios . . . . .	6
<b>3 Análise de Requisitos e Proposta de Arquitetura</b>	<b>7</b>
3.1 Requisitos de Sistema . . . . .	7
3.1.1 Levantamento de requisitos . . . . .	7
3.1.2 Atores . . . . .	8
3.1.3 Casos de Utilização . . . . .	9
3.1.4 Requisitos Funcionais . . . . .	11
3.1.5 Requisitos Não Funcionais . . . . .	12
3.2 Arquitetura do Sistema . . . . .	12
3.2.1 Modelo de Domínio . . . . .	13
3.2.2 Modelo Tecnológico . . . . .	13
3.2.3 Modelo de Instalação . . . . .	14
<b>4 Avaliação Experimental de Tecnologias</b>	<b>15</b>
4.1 Abordagem ao Reconhecimento por <i>Machine Learning</i> . . . . .	15
4.2 Reconhecimento de Comida . . . . .	15
4.3 Reconhecimento de Exercícios Físicos . . . . .	16
4.4 Integração de Fitness Trackers Externos . . . . .	16
<b>5 Implementação</b>	<b>19</b>
5.1 Modelo de Domínio e Base de Dados . . . . .	19
5.2 REST API . . . . .	19
5.2.1 Queries, Django ORM e Bases de Dados . . . . .	19
5.2.2 Autenticação e Autorização . . . . .	20
5.2.3 Reconhecimento de Comida . . . . .	22
5.2.4 Identificação por Código de Barras . . . . .	22
5.2.5 Integração com a Fitbit . . . . .	23
5.2.6 Envio Automático de Notificações . . . . .	23
5.3 Plataforma Web do Cliente/Paciente . . . . .	24

5.3.1	Login e Registo . . . . .	24
5.3.2	Dashboard . . . . .	26
5.3.3	Perfil . . . . .	26
5.3.4	Informação nutricional . . . . .	27
5.3.5	Gestão de <i>food logs</i> . . . . .	28
5.3.6	Médico associado . . . . .	30
5.4	Plataforma Móvel do Cliente/Paciente . . . . .	30
5.4.1	Perfil . . . . .	30
5.4.2	Estatísticas . . . . .	32
5.4.3	Visualização de <i>food logs</i> . . . . .	33
5.4.4	Inserção de <i>food logs</i> . . . . .	34
5.5	Plataforma do Médico . . . . .	36
5.5.1	Login e Dashboard . . . . .	36
5.5.2	Perfil . . . . .	36
5.5.3	Gestão de pacientes . . . . .	37
5.6	Plataforma do Administrador . . . . .	40
5.6.1	Login e Dashboard . . . . .	40
5.6.2	Perfil do administrador . . . . .	40
5.6.3	Gestão de médicos . . . . .	41
<b>6</b>	<b>Discussão de Resultados</b>	<b>43</b>
<b>7</b>	<b>Gestão do Projeto</b>	<b>45</b>
<b>8</b>	<b>Conclusões e Trabalho Futuro</b>	<b>47</b>
	<b>Referências</b>	<b>49</b>

## **Lista de Figuras**

1	Matriz de Confusão associada ao modelo. Fonte: [1] . . . . .	5
2	Modelo de casos de uso. . . . .	9
3	Modelo de domínio . . . . .	13
4	Modelo Tecnológico . . . . .	14
5	Modelo de Instalação . . . . .	14
6	Formulário de registo na aplicação web . . . . .	25
7	Autenticação na aplicação web . . . . .	25
8	Dashboard do utilizador da aplicação web . . . . .	26
9	Página de perfil de um utilizador . . . . .	27
10	Gráficos representativos da métrica MyLife e do batimento cardíaco . . . . .	28
11	Gráficos com a evolução temporal de diversas métricas vitais e nutricionais	28
12	Inserção de um novo <i>food log</i> . . . . .	29
13	Visualização de <i>food log</i> . . . . .	29
14	Detalhes do médico . . . . .	30
15	Página de métrica MyLife . . . . .	31
16	Página de perfil . . . . .	31
17	Página de Body Stats, com as várias opções disponíveis . . . . .	32
18	Página de Heart Rate Stats . . . . .	33
19	Página de um food log . . . . .	34
20	Página de Inserção de um novo food log . . . . .	35
21	Página de Dashboard de um médico . . . . .	36
22	Página de perfil de um médico . . . . .	37
23	Lista de pacientes associados a um médico . . . . .	37
24	Confirmação da remoção de um paciente . . . . .	38
25	Gráficos representativos da métrica MyLife e do batimento cardíaco de um paciente . . . . .	38
26	Gráficos com a evolução temporal de diversas métricas vitais e nutricionais de um paciente . . . . .	39
27	Adição de um novo paciente à lista de pacientes associados . . . . .	39
28	Dashboard do administrador . . . . .	40
29	Página de perfil do administrador do hospital . . . . .	41
30	Lista de médicos associados a um hospital . . . . .	41
31	Formulário de registo de um novo médico . . . . .	42
32	Exemplo de pipeline de integração e deployment . . . . .	45
33	Exemplo de discussão de código em equipa . . . . .	46



## Abreviaturas

<b>CI</b>	Integração Contínua
<b>CD</b>	<i>Continuous Deployment</i>
<b>ML</b>	<i>Machine Learning</i>
<b>CNN</b>	<i>Convolutional Neural Networks</i>
<b>YOLO</b>	<i>Framework You Only Look Once</i>
<b>REST</b>	<i>Representational State Transfer</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>ORM</b>	<i>Object Relational Mapping</i>
<b>DRF</b>	<i>Django Rest Framework</i>
<b>VM</b>	<i>Virtual Machine</i>
<b>Princípios ACID</b>	Atomicidade, Consistência, Isolamento e Durabilidade
<b>BD</b>	Base de Dados



# 1 Introdução

A utilização das tecnologias de informação e comunicação na área da saúde está a ser explorada nos mais diversos cenários de diagnóstico, monitorização e tratamento de pessoas. Na última década, assistiu-se a uma proliferação de dispositivos móveis para monitorização de sinais vitais e atividade física dos cidadãos, fomentando assim o desenvolvimento de um novo paradigma de monitorização de condições clínicas e hábitos comportamentais. Por outro lado, sabe-se que os padrões de consumo alimentar têm um grande impacto nas condições de saúde e, por isso, devem ser monitorizados e acompanhados regularmente.

Assim, este projeto apresenta um meio simples, rápido e intuitivo de registar toda esta informação e utilizá-la para criar métricas e estatísticas que ajudem os utilizadores a perceber se os seus comportamentos são ou não os mais adequados e onde podem melhorar.

## 1.1 Contexto

Este projeto foi desenvolvido sob o tema Projeto em Informática, supervisionado pelos professores António Sousa, José Moreira e Rui Aguiar e orientado pelo professor Carlos Costa.

O seu principal foco foi a criação de um sistema simples e moderno que permitisse facilmente introduzir e registar hábitos do dia-a-dia das pessoas, com especial foco na alimentação, e gerar valor com essa informação. Foi desenvolvido no DETI / IEETA ao longo do segundo semestre do ano letivo de 2019/2020.

## 1.2 Motivação

Devido à crescente exigência do mercado de trabalho e ao facto de que as tarefas do dia-a-dia preenchem cada vez mais a agenda dos cidadãos, bem como outras atividades do nosso quotidiano, as pessoas têm cada vez menos tempo para monitorizar a sua saúde e realizar atividades que promovam um estilo de vida saudável.

Atualmente, existem várias tecnologias no mercado que tentam ajudar as pessoas nesta área, fornecendo ferramentas onde é possível inserir os alimentos consumidos diariamente. Contudo, os seus mecanismos recorrem, essencialmente, à inserção manual da informação, o que é um processo lento e exaustivo.

Para além disso, ginásios e *personal trainers* tentam incutir uma cultura de exercício e um estilo de vida saudável nos cidadãos, mas os preços e o alto número de pessoas que frequentam estes espaços acabam por desencorajar muitos cidadãos a praticar este tipo de atividades.

Posto isto, a utilização de tecnologias como reconhecimento de comida por códigos de barras ou imagem, bem como a integração de *fitness trackers* no sistema, apresenta uma alternativa mais rápida, completa e inovadora, relativamente ao que existe atualmente no mercado.

## **1.3 Objetivos**

O principal objetivo é desenvolver um sistema moderno e multi-plataforma que, de uma forma simples e intuitiva, permita recolher e agregar informação relativa aos hábitos diários dos utilizadores. Este sistema inclui integrações com tecnologias externas como *fitness trackers* e códigos de barras, bem como a utilização de *push-notifications* para garantir a correta recolha de informação e interação com o sistema por parte dos utilizadores. Toda a recolha de informação será utilizada para produzir *feedback* útil e que permita aos utilizadores ter uma percepção da qualidade dos seus hábitos de saúde diários, tal como a qualidade da sua alimentação e atividade física.

## **1.4 Estrutura do Documento**

O presente relatório irá seguir a seguinte estrutura:

- A Secção 2 irá explorar e apresentar o estado de arte relativamente a processos de computação visual, no que toca a reconhecimento de comida e atividade física, bem como a trabalho relacionado na área de fitness.
- A Secção 3 aborda a especificação do sistema, bem como uma visão geral de como o sistema está construído, tanto ao nível das tecnologias como da instalação.
- A Secção 4 demonstra qual foi a nossa abordagem para testar e escolher algumas das tecnologias usadas, como o reconhecimento por *machine learning*.
- A Secção 5 fornece uma vista sobre a implementação do sistema, onde detalhamos a abordagem e implementação de funcionalidades fulcrais no nosso sistema, bem como a justificação da respetiva abordagem.
- A Secção 6 serve para apresentação dos resultados da implementação final, bem como a discussão e comparação destes com tecnologias e produtos existentes.
- A Secção 7 destaca quais os processos utilizados durante o desenvolvimento do projeto, em vista a facilitar toda a implementação do mesmo, tendo em conta o contexto remoto em que o projeto foi desenvolvido.
- Por fim, a Secção 8, descreve algumas conclusões que retirámos do desenvolvimento do projeto, bem como algumas considerações sobre possíveis trabalhos futuros.

## 2 Estado da Arte e Trabalho Relacionado

Tendo definido o conceito para o sistema *MyLife*, procedemos à realização de pesquisa de mercado e da literatura no sentido de conhecer o estado de desenvolvimento nas áreas que procurávamos trabalhar. Com este estudo, tentámos perceber as mais valias e problemas dos atuais sistemas, integrando, na medida do possível, os avanços do estado de arte e evitando opções menos recomendadas. Considerando a nossa visão inicial para o projeto, na qual exploramos as áreas de **reconhecimento de comida**, bem como de **reconhecimento de exercícios**, conduzimos parte da nossa análise inicial nestas duas direções.

### 2.1 Monitorização de Hábitos de Saúde e Exercício Físico

Para nos contextualizarmos com o mercado, encontrámos aplicações que se encaixavam na mesma secção que a nossa. Ao que concluímos que as melhores referências seriam:

- **MyFitnessPal**
- **Freeletics**
- **Fitbit**
- **Google Fit**

A aplicação **MyFitnessPal** enquadra-se bem no nosso problema, pois permite ao seu utilizador registar o seu consumo diário e fornece-lhe ainda feedback consoante esse consumo. É ainda um bom exemplo do que pretendemos atingir, pois é bastante **user-friendly** e tem poucos problemas de usabilidade num cenário quotidiano. Esta permite adicionalmente fazer **tracking dos nutrientes diários** e oferece dados estatísticos relevantes ao utilizador, para que ele consiga atingir as suas metas de perda/ganho de peso. Para além disso, inclui também uma base de dados de alimentos bastante rica e completa, que os utilizadores podem usar para criar as suas próprias refeições, assim como um **leitor de códigos de barras**, para mais facilmente identificar alimentos que os utilizadores tenham à mão.

Quanto ao **Freeletics**, a aplicação dedica-se a fazer treinos de bodyweight personalizados que ajudam o utilizador a atingir as suas metas de condição física, através da personalização do tipo de treino e com recompensas conforme o tempo em que os utilizadores demoram a terminar. A aplicação é limitada nos objetivos que pretendemos do **MyLife**, pois apenas permite fazer treinos personalizados e receber feedback dos mesmos, deixando de lado a informação relativa à alimentação.

A **Fitbit** é um sistema que permite monitorizar atividade física, exercício, batimento cardíaco entre outros. Este está associado a, pelo menos, um *smartwatch* que funciona como um dispositivo de **fitness tracking**, pelo que é bastante importante no desenvolvimento da maioria das funcionalidades do produto. Devido às limitações de sincronização dos dados registados com a sua API, estes não são atualizados em tempo-real e necessitam de calendarização de sincronizações.

A **Google Fit** é uma aplicação de monitorização muito semelhante ao módulo de avaliação de métricas da **Fitbit**, com registo de métricas como: Atividade física, exercício, etc... Esta aplicação "gamifica" o bem-estar físico dos utilizadores através de um sistema de pontos. A pontuação de cada utilizador vai aumentando conforme o exercício que é

realizado, o que leva a que os utilizadores tenham outra motivação para realizar atividade física. Esta aplicação tem também integração com o **MyFitnessPal**, o que é relevante, pois, também nos permite ter então o módulo de análise das ingestões diárias. Porém, visto que se trata apenas de uma aplicação e não de um sistema com dispositivo usável, não permite fazer diretamente o tracking do batimento cardíaco, que é uma métrica muito usada no **MyLife**.

## 2.2 Reconhecimento e Classificação de Comida

No campo do **reconhecimento de comida**, existe, atualmente, algum trabalho desenvolvido. Na área da construção de modelos de *Machine Learning* para o efeito, têm sido desenvolvidos vários trabalhos, dos quais se destaca o conduzido por *David J. Attokaren et. al.*, em [1], onde aborda o problema com recurso a *Convolutional Neural Networks*<sup>1</sup>, uma base de dados de refeições bastante extensa (com cerca de 101 000 imagens divididas em 101 categorias) e um modelo neural pré-treinado (*Inception V3* da Google<sup>2</sup>). Tendo por base estes componentes, a equipa de investigadores conseguiu resultados bastante aceitáveis, como podemos ver na tabela 1.

Sl.No.	Model	Dataset	Accuracy (%)
1	SVM	Food-101	50.76
2	Neural Networks	Food-101	56.40
3	RFDC-based Approach	Food-101	56.76
4	Resnet 18	Food-101	67.23
5	CNN	UEC-FOOD100	78.77
6	CNN (ILSVRC)	Food-101	79.20
7	CNN (Food-101)	EgocentricFood	90.90
8	Proposed Approach	Food-101	<b>86.97</b>

Tabela 1: Comparação de Resultados. Fonte: [1]

Como fica evidente através destes resultados, foi possível obter uma precisão bastante aceitável no que diz respeito a este problema em particular, especialmente quando comparado com outras tentativas.

Por outro lado, do ponto de vista de uma implementação mais prática, uma das referências no campo do reconhecimento de comida é o trabalho realizado por *Patrick Rodriguez*, presente em [2] no qual, seguindo uma estratégia semelhante, partindo do mesmo dataset e modelo base usado por *David J. Attokaren et. al.*, e recorrendo às bibliotecas de *machine learning*: *Tensorflow* e *Keras*, atingiu resultados bastante aceitáveis para uma aplicação deste âmbito, tal como se pode ver na figura 1.

<sup>1</sup>Convolutional Neural Network (CNN): um tipo de rede neural profunda, aplicada principalmente ao reconhecimento de imagens.

<sup>2</sup>Fonte: [https://www.tensorflow.org/api\\_docs/python/tf/keras/applications/InceptionV3](https://www.tensorflow.org/api_docs/python/tf/keras/applications/InceptionV3)

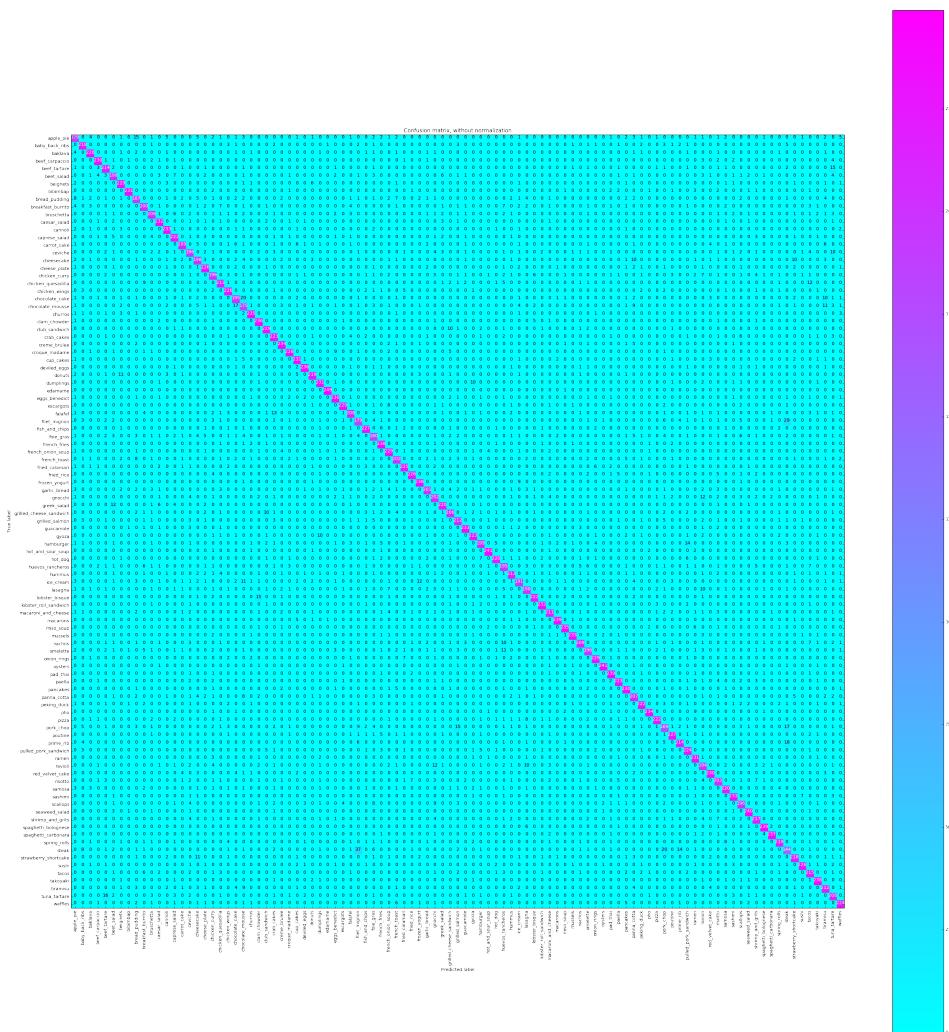


Figura 1: Matriz de Confusão associada ao modelo. Fonte: [1]

Conforme podemos observar na figura 1, a matriz de confusão<sup>3</sup> permite ver que a classificação obtida pelo modelo referido atingiu níveis de exatidão bastante elevados, providenciando uma identificação aceitável para todas as classes envolvidas.

Assim, em conclusão, estes dois projetos mostraram ser duas bases fiáveis para o âmbito do nosso projeto.

---

<sup>3</sup>Matriz de Confusão: representação em forma de tabela e/ou imagem que permite visualizar a performance de um algoritmo, tendo em conta o número, falsos positivos, falsos negativas e classificações corretas.

## 2.3 Reconhecimento de Exercícios

No que diz respeito ao **reconhecimento de exercícios**, existe, igualmente, algum trabalho desenvolvido neste sentido. Nomeadamente, um dos projetos que, à primeira vista, apresenta elevada viabilidade é o *OpenPose* [3]. Este é um projeto *Open Source* que, aplicando um modelo de *Machine Learning* pré-treinado, permite detetar, em tempo real, movimentos chave do corpo humano.

Partindo deste projeto como base, *Steven Chen* e *Richard Yang*, da Universidade de *Stanford*, publicaram o trabalho presente em [4], no qual apresentaram um sistema que permite detectar os movimentos nos exercícios dos utilizadores e gerar recomendações detalhadas e personalizadas acerca de como o utilizador pode melhorar a sua forma física. A sua solução foi testada e validada com base em vídeos dos mesmos a realizar exercícios físicos tanto correta como erradamente.

### 3 Análise de Requisitos e Proposta de Arquitetura

#### 3.1 Requisitos de Sistema

##### 3.1.1 Levantamento de requisitos

Para que conseguíssemos solucionar da melhor forma possível os problemas identificados anteriormente neste documento, foi necessário realizar um levantamento de requisitos através de várias análises, estudos e entrevistas. Com este esforço, objetivou-se desenvolver uma plataforma robusta, com interesse prático e acessível a um universo alargado de pessoas. Neste sentido, decidimos dividir o nosso processo de levantamento de requisitos em três partes distintas:

**Na primeira parte**, discutimos algumas ideias através de **brainstorms**, em grupo, com o intuito de decidir os objetivos, os limites e o público-alvo do sistema.

**Na segunda parte**, desenvolvemos algum trabalho de pesquisa sobre o estado de arte atual nas áreas da saúde, do exercício físico e da nutrição, onde analisámos diferentes artigos, projetos, estudos e tecnologias. Desta forma, conseguimos ter uma melhor percepção sobre o que funcionou e sobre o que não funcionou em cada solução do estado de arte, podendo deste modo melhorar o nosso processo de levantamento de requisitos.

**Na terceira parte**, e provavelmente a mais importante, tivemos algumas reuniões com o nosso orientador, o Prof. Dr. Carlos Costa. A sua experiência em sistemas de informação, relacionados com a área da saúde, foi essencial para tomarmos algumas decisões importantíssimas sobre o que deveríamos ou não incluir no nosso produto final. Através dos seus conselhos, durante todas as fases de desenvolvimento do projeto, fomos capazes de efetuar um levantamento de requisitos de uma forma contínua e progressiva. De um modo geral, estes foram alguns dos tópicos discutidos em conjunto com o nosso orientador:

- Relativamente ao reconhecimento de comida, através de imagens, isto poderia ser uma das funcionalidades mais complicadas do nosso produto, uma vez que este reconhecimento poderia tornar-se difícil caso a comida estivesse misturada ou empilhada. Assim, numa fase inicial, fomos aconselhados a tentarmos reconhecer determinados alimentos separadamente e, só depois, tentar reconhecer múltiplos ingredientes na mesma imagem.
- Quanto ao reconhecimento de exercícios físicos, e ainda que seja um ramo secundário do nosso produto, é de realçar a sua grande complexidade. Por este motivo, e tendo em conta a nossa pesquisa e avaliação, descritas nas secções 2.3 e 4.3, concluímos que seria dada prioridade ao reconhecimento de comida, descrito no ponto anterior, uma vez que as ferramentas disponíveis para a tarefa de reconhecer exercícios não apresentam maturidade suficiente para permitir uma utilização em tempo real num dispositivo móvel.
- Surgiu ainda um outro tópico, relacionado com a eventual integração de um módulo de análise dedicado a doentes com diabetes, uma vez que esta doença afeta uma percentagem bastante significativa da população mundial. O conselho que nos foi dado foi o de dar prioridade à integração da pulseira da Fitbit Charge 3 no nosso sistema e, após isso, tentar integrar o módulo de monitorização da diabetes. A priorização da integração do sistema com a Fitbit seria essencial, pois, em casos de

diabetes é fundamental controlar os hábitos alimentares da pessoa, bem como os seus hábitos de atividade física.

Tendo em conta todas estas informações, e após ponderar todos os prós e contras, foram tomadas as seguintes decisões gerais:

- Numa fase inicial, iríamos integrar o nosso sistema com a pulseira **Fitbit Charge 3**, não só por ser um dispositivo comercial, acessível ao público em geral, mas também por nos fornecer diversas métricas vitais, em tempo real, do seu utilizador, como por exemplo o batimento cardíaco, a pressão arterial, a frequência respiratória, o número de passos e a distância percorrida num determinado dia, entre outras.
- Para que um utilizador comum pudesse interagir com o nosso sistema, ficou decidido que iríamos desenvolver tanto uma aplicação web, como uma aplicação móvel, em que o utilizador poderia consultar diversas métricas de saúde, físicas e vitais em tempo real, bem como analisar a evolução do seu estado geral de saúde. Através da aplicação móvel, poderia ainda tirar fotografias a diferentes refeições, para que pudessem ser reconhecidas no nosso sistema através de um módulo de aprendizagem automática.
- Numa vertente mais secundária, decidimos também que iríamos desenvolver uma aplicação web para ser utilizada por profissionais de saúde (médicos, enfermeiros, psicólogos, *personal trainers*, entre outros), onde poderiam aceder e consultar o estado geral de saúde de um determinado utilizador, com a devida autorização deste.

### 3.1.2 Atoes

Ao conceber um sistema de informação, é indispensável identificar quais os atores que irão interagir com as interfaces do sistema. Essencialmente, estes atores são entidades representativas de um grupo de utilizadores com características comuns entre si. Neste sentido, também poderão ser definidas *personas*, que são atores específicos com características baseadas em dados reais.

Considerando então a importância da identificação dos atores que irão utilizar o nosso sistema, temos as seguintes situações:

- Primeiramente, o principal público-alvo do nosso sistema é o cidadão comum que esteja interessado em monitorizar o seu estilo de vida de uma forma mais simples e completa, tanto a nível alimentar como a nível físico. Neste sentido, estas pessoas poderão interagir com o nosso sistema, quer através da aplicação móvel, quer através da aplicação web.
- Em segundo lugar, e em situações que façam sentido, diversos médicos associados a um hospital poderão interagir com o nosso sistema através de uma aplicação web, onde poderão adicionar os utilizadores descritos no ponto anterior à sua lista de pacientes e visualizar o estado de saúde destes através de um conjunto de métricas, gráficos e análises temporais. Desta forma, o acompanhamento do paciente por parte do médico será muito mais regular e cuidado, agilizando também as consultas presenciais.
- Por fim, as instituições que utilizam o nosso serviço terão à sua disposição uma interface de administrador, onde poderão gerir os seus médicos e outros profissionais de saúde.

Após estas considerações, os nossos atores ficam então definidos, resumidamente, da seguinte forma:

- **Cliente (eventualmente, também um paciente):** Utilizador com acesso tanto à aplicação móvel bem como à aplicação web. De um modo geral, este ator consegue registar e controlar as refeições que ingere, adicionar um dispositivo Fitbit ao seu perfil e visualizar o seu histórico global de saúde.
- **Médico:** Este ator terá acesso a uma aplicação web, onde poderá gerir a sua lista de pacientes e acompanhar o estado de saúde de cada um.
- **Administrador:** Assim como o médico, este ator terá acesso a uma aplicação web, onde poderá gerir os médicos da sua instituição, bem como consultar estatísticas gerais.

### 3.1.3 Casos de Utilização

No seguimento do ponto anterior, referente à definição dos atores que interagem com o nosso sistema, é importante definir em que cenários é que essas interações são realizadas. Neste sentido, identificámos o conjunto de casos de utilização adequados ao nosso sistema, como podemos verificar no diagrama presente na figura 2.

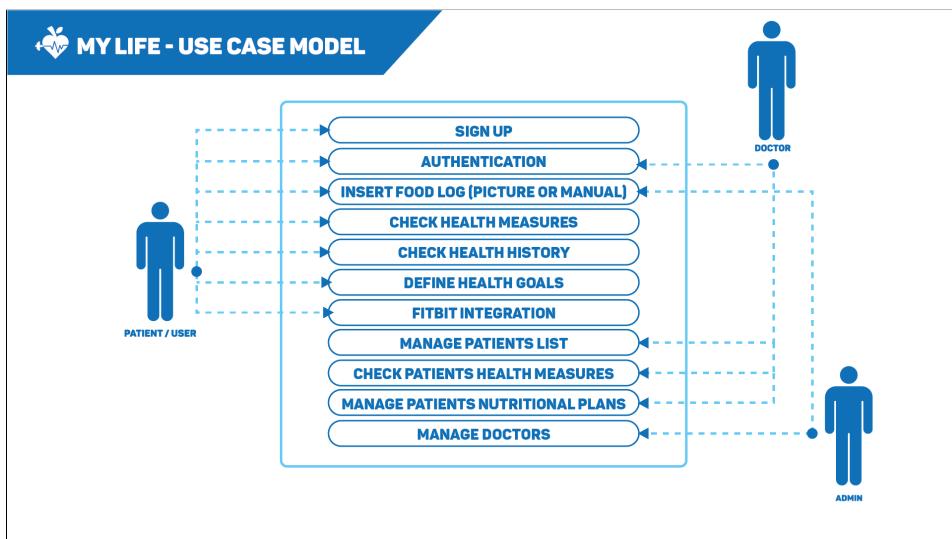


Figura 2: Modelo de casos de uso.

Para detalhar com mais cuidado cada grupo de interação, apresentamos também uma pequena descrição e prioridade de cada um:

#### Casos de utilização do Cliente

À exceção do caso de uso de inserção de food logs, os casos de uso respetivos ao cliente, são tanto aplicáveis na aplicação mobile como na web.

- **Inserção de food logs:** Deve aceder à página de food logs e clicar no botão de adicionar food log (no mobile este botão está presente em todos os ecrãs na barra de navegação inferior), e proceder com a inserção de alimentos. No caso de inserção manual (comidas já existentes), o processo é resumido no preenchimento de um formulário. Caso pretenda inserir através de reconhecimento de comida, existem formas diferentes conforme a plataforma:

- **Web:** Para inserir na web, basta inserirmos uma fotografia que já esteja no computador.
- **Mobile:** No mobile, podemos escolher uma fotografia já presente na galeria, tirar uma fotografia nesse instante ou, reconhecer através de código de barras.

**Prioridade:**Alta

- **Verificar métricas de saúde:** O cliente quer verificar as suas métricas de saúde. O cliente vai à área pessoal onde consegue verificar as métricas ou, tem a oportunidade de ir à página de estatísticas e verificar os consumos dos macronutrientes e o seu histórico de ritmo cardíaco.

**Prioridade:**Alta

- **Verificar histórico nutricional:** O cliente quer ter acesso ao histórico de refeições ingeridas. Para isso, dirige-se à página de food logs, acessível através da barra inferior e vai se deparar com o dia atual. Caso pretenda verificar os dias anteriores e posteriores, pode fazê-lo, clicando nas setas.

**Prioridade:**Média

- **Definir objetivos de peso:** O cliente quer definir um objetivo de peso ideal, e a partir daí, quer que o sistema lhe indique quais as quantidades diárias dos nutrientes. Para isso, dirige-se à sua página de perfil e edita o seu peso objetivo.

**Prioridade:**Média

- **Integração com Fitbit (Apenas Mobile):** O cliente quer ligar a sua Fitbit à aplicação móvel para poder extrair dados importantes para as suas métricas, como passos, batimento cardíaco, entre outras. Para tal, e caso ainda não tenha feito a ligação, dirige-se à página de perfil e clica no botão “Connect your Fitbit”, e segue os passos de autenticação da FitBit.

**Prioridade:**Alta

## Casos de utilização do Médico

- **Gerir pacientes:** Talvez uma das funcionalidades mais importantes do médico. Para fazer uso de todas as outras funcionalidades, o médico tem que adicionar os seus pacientes à plataforma, de forma a os poder monitorizar.

**Prioridade:**Alta

- **Ver métricas de saúde:** Quando o médico quer verificar o estado de saúde de um dos seus pacientes, primeiro tem que se dirigir à respetiva área no dashboard. Assim que se encontre na página vai conseguir ver a listagem dos seus pacientes e pode escolher qualquer um deles. Depois destes passos, a aplicação vai-lhe indicar as métricas do paciente e uma opção para o contactar.

**Prioridade:**Alta

- **Gerir planos nutricionais dos pacientes:** Para cada paciente, o médico consegue ver e gerir o plano nutricional, ou até fazer recomendações. O médico consegue editar detalhes específicos como por exemplo calorias ou ingredientes de cada refeição. Para além disto, consegue visualizar todos os food logs inseridos pelo cliente

**Prioridade:**Alta

### Casos de utilização do Administrador do hospital

- **Gerir doutores:** O administrador consegue ver a listagem de médicos associados a um hospital, e tem a possibilidade de os editar se necessário. Assim como a funcionalidade de os adicionar ou remover.

**Prioridade:**Alta

#### 3.1.4 Requisitos Funcionais

Em qualquer projeto de software é fundamental identificar os requisitos funcionais do sistema. Neste sentido, e numa fase inicial do projeto, identificámos os seguintes requisitos funcionais, relacionados com o nosso sistema de informação:

##### 1. Regras de negócio

- Separação entre os utilizadores, os médicos e os administradores, tanto na lógica de negócio como nas aplicações dos clientes.
- Ligação simples e intuitiva entre todas as funções de negócio.

##### 2. Funções administrativas

- O ator do hospital desempenha funções administrativas.
- O ator do médico desempenha um papel administrativo na lógica de negócio.

##### 3. Níveis de autenticação e autorização

- Todos os atores devem estar autenticados para acederem às respetivas funcionalidades.
- São atribuídas permissões específicas a cada utilizador conforme o seu papel.

##### 4. Relatórios

- O sistema deve reportar, automaticamente, erros e paragens à equipa de desenvolvimento.
- Os utilizadores devem poder reportar erros ou problemas.

## 5. Informação histórica

- Eventualmente, suportar a integração com serviços *legacy*.

## 6. Requisitos regulamentares

Considerando os casos de utilização da secção 3.1.3, tem-se que:

- A informação gerada pelo utilizador deve estar disponível através de pelo menos uma plataforma (em concreto, na aplicação web ou na aplicação móvel).
- A informação gerada pelo paciente deve estar disponível para o médico, através da lógica de negócio e persistência na base de dados.
- O médico, através da sua interface, deve poder pesquisar por um determinado utilizador, registado no sistema, e adicioná-lo à sua lista de pacientes.
- O utilizador deve poder fotografar diversos alimentos e adicionar a informação nutricional associada ao sistema.
- Adicionalmente, o utilizador também poderá inserir, manualmente, diversas informações nutricionais, quer através da aplicação móvel, quer através da aplicação web.

### 3.1.5 Requisitos Não Funcionais

A seguinte lista apresenta os requisitos não funcionais essenciais para garantir a qualidade do nosso projeto:

- **Usabilidade:** Dadas as características de uma aplicação móvel e a necessidade da utilização partir do user, era necessário tornar a aplicação fácil de aprender e de usar. Portanto fizemos vários esforços neste sentido, ao reduzir o número de interações, facilidade de navegação, etc..
- **Performance:** Visto que pretendemos alcançar um público-alvo bastante extenso e não queremos que o uso da aplicação seja um dispender de tempo enorme no quotidiano dos mesmos, foi muito importante tornar a aplicação usável em vários dispositivos e que tivesse boa performance.
- **Recuperabilidade:** Visto que a aplicação pega em dados inseridos pelo utilizador e há sempre margem de erro, foi necessário tornar a aplicação recuperável, na medida em que permita o utilizador apagar/retroceder, caso tenha inserido alguma medida errada.
- **Interoperabilidade:** Devido à junção das métricas obtidas através da Fitbit, era necessário haver esta interoperabilidade, o que nos permitiu criar aplicações Web/Android/iOS sem configurações necessárias.

## 3.2 Arquitetura do Sistema

O sistema possui 3 componentes principais: a **aplicação web**, a **aplicação móvel** e a **REST API**, consumida pelos dois componentes anteriores. É importante referir que a segurança dos dados dos nossos clientes é algo que tivemos em conta quando desenhámos o nosso sistema. Os dados dos utilizadores seguem um conceito de Avatar em que as informações

disponibilizadas pelos utilizadores podem não ser reais, logo estarão a criar uma personagem fictícia dentro do nosso sistema.

Além disso, gostaríamos de no futuro aplicar uma anonimização mais restrita sobre os dados do nosso projeto, encriptando comunicações entre serviços, encriptando informação sensível entre outras medidas.

### 3.2.1 Modelo de Domínio

No diagrama do modelo de domínio, descrevemos as várias relações entre as entidades existentes no sistema. Esta modelação foi essencial para conseguirmos programar mais facilmente as relações entre as diferentes entidades do sistema. Como podemos inferir através da figura 3 existem 3 utilizadores principais no nosso projeto: o **Doctor**, o **Client** e o **Admin**. O **Client** pode estar associado a um **Doctor** e além disso pode ter associado um conjunto de históricos de refeições que são caracterizados no sistema pela entidade **Meal History**.

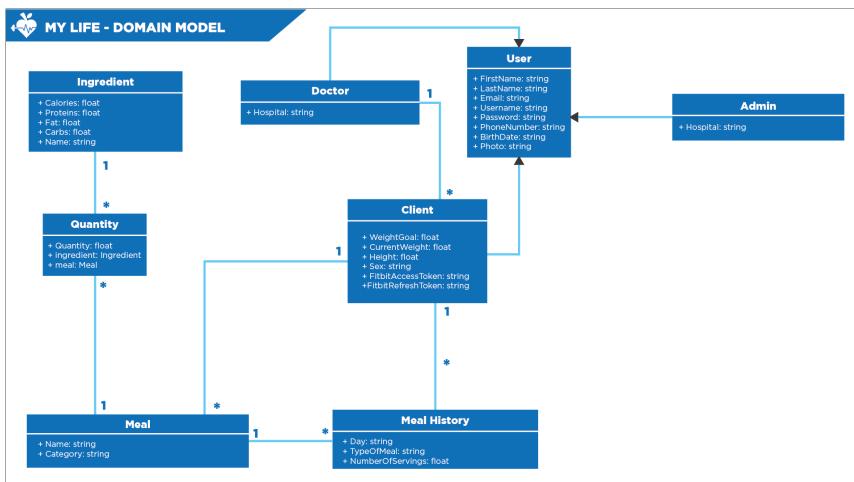


Figura 3: Modelo de domínio

### 3.2.2 Modelo Tecnológico

Para a aplicação web utilizámos a biblioteca de JavaScript **ReactJS** e para a aplicação móvel utilizámos **React Native**, garantindo uma maior portabilidade de componentes entre as duas plataformas.

Para o desenvolvimento da REST API utilizámos a framework **Django** e **Django REST Framework** (baseada em Python), que permite um desenvolvimento rápido, escalável e versátil, características essenciais para o desenvolvimento do nosso projeto.

Para adicionar persistência ao sistema utilizámos uma base de dados **PostgreSQL**, como podemos observar na figura 4, devido ao facto de ser um sistema de gestão de bases de dados *Open Source* e seguir os princípios ACID (Atomicidade, Consistência, Isolamento e Durabilidade).

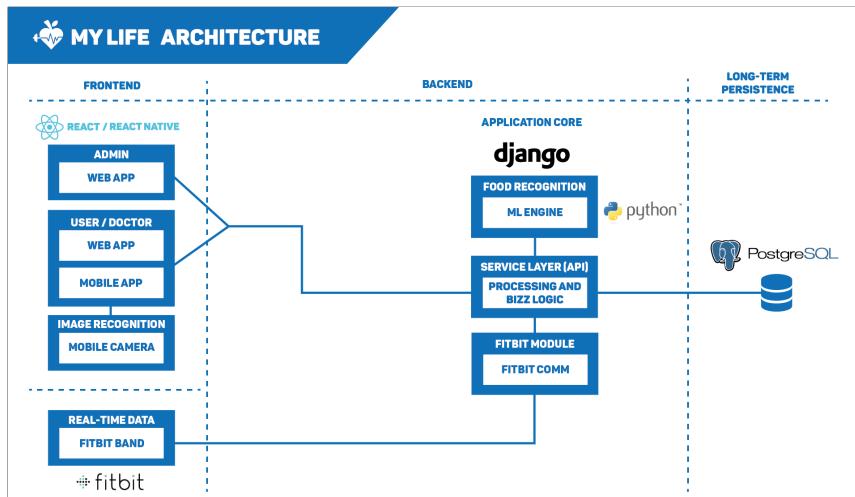


Figura 4: Modelo Tecnológico

### 3.2.3 Modelo de Instalação

No diagrama de instalação apresentado em baixo, figura 5, podemos verificar que a aplicação web do utilizador, médico e administrador está hospedada num servidor web e que a aplicação móvel é instalada no telemóvel do utilizador. A REST API, o módulo de reconhecimento de ML e a Base de dados relacional estão armazenados no mesmo servidor, sendo que a REST API é o único ponto de ligação direto com as aplicações web e móvel.

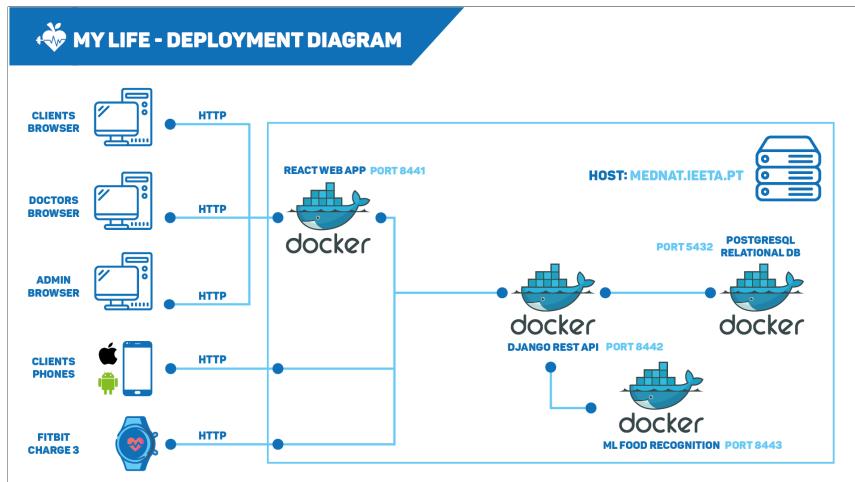


Figura 5: Modelo de Instalação

## 4 Avaliação Experimental de Tecnologias

### 4.1 Abordagem ao Reconhecimento por *Machine Learning*

Tendo em conta o trabalho existente que encontrámos no campo de *Machine Learning*, optámos por direcionar a nossa pesquisa para a reutilização e adaptação de tecnologias desenvolvidas e testadas.

O nosso objetivo principal era obter um produto como prova de conceito utilizável, pelo que conduzimos a nossa pesquisa no sentido de obter uma solução viável num contexto real.

Nesse sentido, algo que, desde o início, estabelecemos como base, foi o requisito de que a nossa solução não poderia envolver um serviço que limitasse ou compromettesse a viabilidade do produto, fosse devido ao desempenho, estabilidade, fiabilidade ou viabilidade.

Por outro lado, e seguindo várias sugestões apontadas pelo Professor Doutor Carlos Costa, estabelecemos objetivos-base mais realistas. Sabendo a complexidade do desafio, assumimos o objetivo de providenciar um serviço que, não sendo perfeito, fosse capaz de produzir uma aproximação à realidade mais precisa que a estimativa do utilizador, tanto no campo de reconhecimento de comidas, como na análise de exercício físico.

Finalmente, e dados estes requisitos pré-estabelecidos, a nossa abordagem ao desenvolvimento dos componentes envolvendo *Machine Learning* e Computação Visual envolveu uma forte componente de teste prático, para assegurar a viabilidade da solução.

### 4.2 Reconhecimento de Comida

Com respeito ao reconhecimento de comida, conduzimos a nossa pesquisa no sentido da construção de um sistema que permitisse o reconhecimento e análise **em tempo real** de comida, através de imagens.

Além dos referidos anteriormente, um dos requisitos que estabelecemos relativamente a este módulo foi a necessidade de este estar isolado e ser executado num servidor *linux*, com elevada disponibilidade, e não no dispositivo cliente da aplicação, fosse num browser ou numa aplicação. Desta forma, poderíamos assegurar a portabilidade do nosso sistema, o isolamento entre os diferentes módulos, e responderíamos aos problemas consequentes de tentar correr modelos de *Machine Learning* complexos num dispositivo móvel, tais como: o baixo nível de recursos no mesmo, a grande variedade de sistemas aos quais teríamos que adaptar o nosso programa e a dificuldade em fazer alterações e melhoramentos ao modelo.

Nesse sentido, iniciámos o desenvolvimento deste módulo em diferentes direções, sendo que a principal partia do trabalho realizado por *Patrick Rodriguez*, em [2], devido à elevada precisão que apresentava dado o problema que pretendíamos resolver, tal como detalhámos na secção 2.2. Após uma extensa pesquisa relacionada com os conceitos de *Machine Learning* que precisávamos, percebemos a necessidade de customizar e adaptar o mesmo ao nosso problema. Ao longo da pesquisa e desenvolvimento, encontrámos problemas como a **dificuldade a nível de recursos computacionais em treinar um modelo** deste género e a necessidade de adaptar o dataset à nossa situação em particular.

Tendo explorado completamente a possibilidade de treinar o nosso próprio modelo, e após várias consultas com o nosso orientador, o Professor Doutor Carlos Costa, bem

como com a Professora Doutora Pétia Georgieva, adaptamos o nosso rumo no sentido de aproveitar um modelo de *Machine Learning* já treinado e funcional para o nosso sistema.

Assim, conduzimos a nossa pesquisa no sentido da utilização da framework YOLO [5], um sistema moderno de reconhecimento de objetos em tempo real, que permite reconhecimento rápido e preciso. Dentro desta solução para o problema, pudemos conduzir a nossa pesquisa ao encontro do trabalho realizado por *Benny Cheung*, disponível em [6], no qual construiu um modelo de *Machine Learning* baseado em YOLO para reconhecimento de comida. Tendo estabelecido esta como a solução mais realisticamente atingível para o nosso caso, procedemos a mais alguma pesquisa neste âmbito tendo, finalmente, entrado em contacto com *Benny Cheung* para obter mais informações e ajuda neste sentido. Dessa maneira, conseguimos aproveitar o seu trabalho e adaptá-lo para a solução do nosso problema.

### 4.3 Reconhecimento de Exercícios Físicos

Com respeito ao reconhecimento de exercícios, conduzimos a nossa pesquisa no sentido da construção de um sistema que permitisse o reconhecimento e análise **em tempo real** da realização de exercícios físicos.

Com isto em mente, e tendo em conta a pesquisa de tecnologias realizada e detalhada na análise do estado de arte, secção 2.3, baseámos o nosso desenvolvimento ao redor do módulo *OpenPose*, com detalhe em [3].

No entanto, ao realizar alguma pesquisa e testes, rapidamente nos deparamos com problemas que poderiam afetar a nossa solução. Em primeiro lugar, assim como no módulo de reconhecimento de comida, surgiu o problema do **tempo e recursos necessários para treinar o modelo** de *Machine Learning*. Por outro lado, dada a elevada complexidade do problema, deparamo-nos com a dificuldade em **executar a solução num dispositivo móvel**, o que seria necessário devido ao facto de ser um sistema que pressupõe *feedback* instantâneo. Finalmente, um dos principais problemas, e que é, simultaneamente, um dos mais comuns neste campo, é a limitação relativa ao tempo de resposta do sistema num reconhecimento em tempo real, o que introduziria **latência** em toda a solução.

Deste modo, dadas todas estas circunstâncias, tendo em conta os requisitos que havíamos previamente estabelecido, e após diversas consultas com o orientador do projeto, o Professor Doutor Carlos Costa, optámos por não implementar esta funcionalidade, por concluirmos que as ferramentas disponíveis não garantiam os nossos requisitos, dando, por outro lado, mais ênfase às restantes componentes.

Conseguimos, assim, garantir não só um produto **funcional e focado**, mas também potenciar uma **melhor usabilidade** com as restantes funcionalidades propostas.

### 4.4 Integração de Fitness Trackers Externos

Um dos pilares do nosso produto é a possibilidade de conectar com *fitness trackers* externos, possibilitando a análise de métricas de exercício físico em tempo real e a consulta do respetivo histórico.

Inicialmente pesquisámos sobre a melhor forma de integrar um *fitness tracker* da *Fitbit* com o nosso sistema, chegando a um plano inicial que envolvia o uso de uma base de dados *time series* que suportaria a persistência dos dados e um *message broker* que serviria de intermediário entre a *Fitbit* e a base de dados.

Após uma análise mais detalhada, chegámos a uma solução mais intuitiva que envolve contactarmos uma API fornecida pela *Fitbit* [7], que permite recolher informação relativa ao batimento cardíaco, distância percorrida, entre outras métricas. Com esta solução apenas precisamos de armazenar um *token* de autenticação, fornecido pela *Fitbit*, que identifica unicamente cada utilizador e, usar o respetivo *Auth Token* sempre que pretendemos consultar informação sobre um utilizador específico.

Para além desta abordagem ser mais acessível, também nos retira alguma da responsabilidade na gestão dos dados do utilizador, pois, toda a informação mais sensível relacionada com a monitorização dos hábitos do cliente é gerida e armazenada pela *Fitbit*.



## 5 Implementação

### 5.1 Modelo de Domínio e Base de Dados

Tal como demonstrado na figura 3, o Modelo de Domínio representa inequivocamente a estrutura lógica de relações no nosso sistema e foi a base para o desenho da Base de Dados.

A Base de Dados (BD) utilizada, serve para guardar toda a informação crucial ao funcionamento da nossa plataforma e foi implementada em PostgreSQL. A sua implementação é gerada através dos *Django Models* como se pode perceber na secção 5.2. O hosting da mesma foi feito com o recurso a docker-compose através do mecanismo de volume na nossa VM.

### 5.2 REST API

Em relação à REST API, esta foi construída com recurso à framework Django, mais especificamente à *DRF (Django Rest Framework)*. Esta permite-nos uma enorme flexibilidade e agilidade, uma vez que é capaz de executar um rápido processamento de requests e facilmente lhe adicionamos novos métodos, que permitem executar novas operações.

Desta framework foi utilizado o seguinte:

- **Django Models:** utilizámos *Django Models* para criar entidades na base de dados, através de ORM (*Object Relational Mapping*).
- **Django Authentication:** tendo por base aos mecanismo de autenticação da DRF, criámos um mecanismo próprio de autenticação baseado em *Auth Tokens*.
- **Django Authorization:** utilizado para controlar o acesso dos utilizadores ao sistema, tendo em conta os privilégios associados à sua conta.
- **Django REST Framework:** utilizado para a criação da REST API.

#### 5.2.1 Queries, Django ORM e Bases de Dados

Através dos recursos providenciados pelo Django, é possível facilmente manipular objetos da Base de Dados que está a ser utilizada pela API. O Django ORM permite-nos mapear entidades da base de dados em objetos de python, que podem, posteriormente, ser manipulados. É possível estabelecer relações entre entidades, bem como criar, remover ou editar qualquer entidade da base de dados.

Para separar a lógica de acesso à base de dados da lógica de “negócio”, decidimos incluir toda a lógica de acesso à BD num ficheiro separado, chamado *queries.py*, onde existem todas as *queries* utilizadas pela REST API.

Como se pode ver no excerto de código 1, um exemplo de uma *query* à base de dados com o objetivo de criar um novo cliente, percebe-se que o acesso e criação de entidades e o consequente mapeamento das mesmas para a base de dados é bastante simples.

---

```
def add_client(data):
    height = data.get("height")
    weight_goal = data.get("weight_goal")
    current_weight = data.get("current_weight")
```

```

sex = data.get("sex")

is_diabetic = data.get("is_diabetic", False)
has_high_colesterol = data.get("has_high_colesterol", False)

state, content = add_user(data)
if not state:
    return state, content

custom_user = content
user = custom_user.auth_user

try:
    # link the user to a client
    Client.objects.create(user=custom_user, height=height,
                          current_weight=current_weight, weight_goal=weight_goal,
                          sex=sex, is_diabetic=is_diabetic,
                          has_high_colesterol=has_high_colesterol)

except Exception:
    user.delete()
    error_message = "Error while creating new client!"
    return False, error_message

# check if the client group exists, else create it
# finally add client to group
try:
    clients_group, created = ...
    clients_group.user_set.add(user)

except Exception:
    user.delete()
    error_message = "Error while creating new client!"
    return False, error_message

state_message = "Client was registered successfully!"
return True, state_message

```

---

Código 1: Função de acesso à BD para criação de um novo cliente

### 5.2.2 Autenticação e Autorização

Tendo em conta a utilização de uma REST API para aceder à informação do nosso sistema, é essencial que os mecanismos de autenticação e autorização existam ao nível da API e não nas diferentes interfaces de utilização. Caso contrário, seria possível obter informação crítica dos nossos utilizadores sem qualquer autorização nem controlo de privilégios de acesso.

Assim, optámos pela utilização de *Auth Tokens*, que devem acompanhar todos os pedidos feitos à REST API, à exceção do login e do registo de clientes.

Inicialmente, os utilizadores enviam os seus dados de autenticação através de um método POST, que, caso a autenticação seja bem sucedida, retorna ao utilizador um *Auth*

*Token*, que este deve guardar para posteriores pedidos. Este *Auth Token* é, também, guardado pelo sistema, de forma a verificar se um utilizador pode, ou não, aceder ao mesmo.

Como pode ser visto no excerto de código 2, quando acaba o tempo de vida do *Auth Token* ou o utilizador sai da plataforma, via *logout*, este é removido do sistema, sendo que, num futuro acesso do mesmo utilizador, um novo *Token* será gerado.

---

```
def expires_in(token):
    time_elapsed = timezone.now() - token.created
    left_time = timedelta(seconds=TOKEN_EXPIRED_SECONDS)
    - time_elapsed
    return left_time

def is_token_expired(token):
    return expires_in(token) < timedelta(seconds=0)

def token_expire_handler(token):
    is_expired = is_token_expired(token)
    if is_expired:
        token.delete()
        token = Token.objects.create(user=token.user)
    return is_expired, token

class ExpiringTokenAuthentication(TokenAuthentication):
    """
    If token is expired then it will be removed
    and new one with different key will be created,
    this new one will be given to the user only the 1st time
    """
    def authenticate_credentials(self, key):
        try:
            token = Token.objects.get(key=key)
        except Token.DoesNotExist:
            raise AuthenticationFailed("Invalid token!")

        if not token.user.is_active:
            raise AuthenticationFailed("Invalid credentials!")

        is_expired, token = token_expire_handler(token)
    return token.user, token
```

---

Código 2: Controlo e validação de *Auth Tokens*, para acesso

Por fim, tendo por base a implementação de *roles* e grupos, como se pode ver excertos de código 3, é na API que é verificada a autorização que um utilizador detém para aceder a determinados recursos.

---

```

def get_role(username, request=None):
    role = None
    try:
        if username is None:
            username = request.user.username

        if User.objects.get(username=username).is_superuser:
            role = "django-admin"
        elif User.objects.get(username=username).groups.all()[0].name in \
                ["admins_group"]:
            role = "admin"
        elif User.objects.get(username=username).groups.all()[0].name in \
                ["clients_group"]:
            role = "client"
        elif User.objects.get(username=username).groups.all()[0].name in \
                ["doctors_group"]:
            role = "doctor"

    except User.DoesNotExist:
        role = None

    return role

```

---

Código 3: Função para identificar e distinguir roles

### 5.2.3 Reconhecimento de Comida

Tendo por base a pesquisa realizada previamente no campo do reconhecimento de comida usando *Machine Learning* 4.2, construímos a nossa solução no backend ao redor do sistema *YOLO* para a análise e classificação de imagens [5].

Este sistema baseia-se, por sua vez, no sistema *Darknet* [8], um motor de construção de redes neurais escrito em C. Com este recurso como base, conseguimos, como referido anteriormente, adaptar o trabalho previamente realizado por *Benny Cheung* [6], e aplicá-lo à nossa solução. Para isso, começámos por usar, efetivamente, *Darknet* para visualizar as previsões em tempo real relativas às imagens de comida que fornecímos como input.

Por outro lado, dado que o nosso conceito inicial pressupunha o reconhecimento como parte de um serviço num servidor *linux*, foi necessário adaptar o módulo *YOLO* à execução sem um sistema gráfico, como o *Darknet*. Nesse sentido, usamos uma sua variante, *Darkflow* [9]. Estando baseada em *Tensorflow*, este sistema permite-nos traduzir e exportar modelos treinados em *Darknet* de forma a poder utilizá-los com código *Python*.

Dado isto, implementámos uma pequena API em *Flask*, com a função de receber, como pedidos, imagens, e responder com a respectiva classificação realizada pelo modelo.

Ao separar todos os nossos módulos em *Docker containers*, esta abordagem permitiu-nos isolar cada componente, mantendo cada módulo leve, eficiente, de fácil manutenção e elevada portabilidade.

### 5.2.4 Identificação por Código de Barras

Para complementar o mecanismo de identificação de comida e, assim, alargar a versatilidade do nosso sistema, implementámos um mecanismo que permite identificar produtos

alimentares através do seu código de barras. Para isso, utilizámos um *dataset* existente online - *Open Food Facts Data* [10] -, com uma grande variedade de produtos, tanto a nível nacional como internacional.

Neste sentido, utilizámos uma REST API externa [11], disponibilizada pelos responsáveis pelo *dataset*, que nos permite consultar toda a informação associada a um dado código de barras. Isto facilitou bastante o desenvolvimento deste componente, pois, retirou a necessidade de ter toda esta informação numa base de dados própria e permite-nos de uma forma bastante simples ter acesso a um conjunto considerável de informação relativa aos mais variados produtos alimentares. Contudo, esta abordagem introduz uma dependência a um serviço externo que tem de ser analisada e monitorizada periodicamente para garantir que a funcionalidade tem o comportamento esperado.

### 5.2.5 Integração com a Fitbit

Uma outra importante vertente do nosso sistema, como já foi apresentada, é a integração com dispositivos de *tracking*. Tal como discutido na secção 4.4, tendo em conta a disponibilidade do nosso orientador Professor Doutor Carlos Costa, e a vasta distribuição e utilização dos dispositivos Fitbit, decidimos incluir a integração da informação recolhida e disponibilizada por estes no nosso sistema. Esta informação é utilizada para cálculo de estatísticas e métricas próprias para melhor enquadrar os hábitos de saúde dos utilizadores.

Esta integração é feita através da comunicação com a REST API externa da Fitbit, à qual se acede através de um *token* próprio que nos é fornecido quando o utilizador decide associar a sua conta Fitbit ao seu perfil MyLife.

A utilização desta abordagem trouxe-nos a vantagem de conseguirmos analisar informação importante com grande fiabilidade e trazer este valor para o nosso sistema, para além de facilitar o acesso à informação, pois, remove a necessidade de guardar qualquer tipo de informação relativa ao histórico de atividade física dos utilizadores, respeitando assim políticas de privacidade e de dados sensíveis.

### 5.2.6 Envio Automático de Notificações

Um dos potenciais problemas que identificámos ao longo do desenvolvimento deste projeto foi o desinteresse ou descuido, por parte dos utilizadores, em criar ou atualizar os *food logs* diariamente. Neste caso, o sistema seria incapaz de processar e analisar a informação nutricional de forma fidedigna, pelo que as recomendações a fornecer ao utilizador seriam, neste modo, imprecisas.

Assim, para combater este potencial problema, decidimos implementar um mecanismo de envio automático de notificações, exclusivamente para a aplicação móvel do utilizador. Essencialmente, este mecanismo foi desenvolvido através de um *cronjob* que executa, periodicamente, um pequeno script em Python. Em concreto, este script é executado duas vezes por dia, às 15 horas e às 21 horas, realizando as operações presentes no algoritmo 1.

---

**Algoritmo 1** Processo de decisão do cronjob de notificações

---

```
usersList ← allUsers
for each user in usersList do
    if not any new food log today then
        devicesList ← user.getDevices()
        for each device in devicesList do
            sendDailyNotification(device)
        end for
    end if
end for
```

---

## 5.3 Plataforma Web do Cliente/Paciente

Como referimos na secção 3.1.3, referente aos casos de utilização de cada ator, o utilizador tem acesso a uma plataforma web, de forma a interagir com o nosso sistema. Esta aplicação web, desenvolvida em ReactJS, está hospedada na nossa máquina no IEETA, podendo ser acedida por diversos clientes. De referir ainda que, ao desenvolvê-la, tivemos muito cuidado em torná-la o mais simples, intuitiva e usável possível, com o intuito de melhorar a experiência do utilizador.

De seguida, iremos descrever as principais funcionalidades desta aplicação web.

### 5.3.1 Login e Registo

Como qualquer plataforma web comum, o nosso sistema permite que qualquer pessoa se registe nele, através do preenchimento de um formulário, de forma a ter acesso a todas as funcionalidades disponíveis. Assim, o utilizador terá de preencher os seguintes campos:

- Primeiro nome.
- Último nome.
- Email.
- Palavra-passe.
- Confirmação da palavra-passe.
- Indicação se é diabético ou se tem o colesterol alto.
- Número de telemóvel.
- Data de nascimento.
- Altura.
- Sexo.
- Peso atual.
- Peso objetivo.

De realçar que estes campos foram corretamente validados, do lado do cliente, com uma mensagem de erro personalizada, para que não seja enviada informação errada para o servidor. De seguida, na figura 6, segue-se uma captura de ecrã ilustrativa deste formulário.

A captura de ecrã mostra um formulário de registo intitulado "Sign Up". O formulário está centralizado em uma janela com fundo azul. No topo, há um ícone de coração com batimentos cardíacos. Abaixo do ícone, o texto "Sign Up" é exibido. O formulário contém os seguintes campos:

- First name \* (campo com placeholder "John")
- Last name \* (campo com placeholder "Doe")
- Email (campo com placeholder "john.doe@example.com")
- Password \* (campo com placeholder "P@ssw0rd")
- Confirm password \* (campo com placeholder "P@ssw0rd")

O botão "NEXT →" está posicionado no fundo do formulário.

Figura 6: Formulário de registo na aplicação web

Caso o registo seja efetuado com sucesso, o utilizador será redirecionado para a página de Login, de forma a autenticar-se no nosso sistema. Assim, após introduzir as suas credenciais de acesso, e destas terem sido corretamente validadas pelo servidor, o utilizador irá ficar autenticado, sendo redirecionado para a Dashboard inicial. De seguida, segue-se uma captura de ecrã, figura 7, do formulário de autenticação:

A captura de ecrã mostra o formulário de login intitulado "Sign In". O formulário está centralizado em uma janela com fundo azul. No topo, há um ícone de coração com batimentos cardíacos. Abaixo do ícone, o texto "Sign In" é exibido. O formulário contém os seguintes campos:

- Email (campo com placeholder "john.doe@example.com")
- Password (campo com placeholder "P@ssw0rd")

O botão "SIGN IN" está posicionado no fundo do formulário.

Figura 7: Autenticação na aplicação web

### 5.3.2 Dashboard

Como referimos na secção anterior, quando um utilizador se autentica na nossa aplicação web, irá ser redirecionado para a *Dashboard* inicial da interface. Nesta página, estão disponíveis, para além da data atual, diversas informações nutricionais do utilizador, tais como:

- Batimento cardíaco.
- Número de passos percorridos, com a respetiva distância em quilómetros.
- A altura e o peso atuais.
- O seu peso objetivo.
- Um gráfico circular, em fatias, com a indicação percentual da quantidade ingerida de cada nutriente (nomeadamente de hidratos de carbono, gorduras, proteínas, entre outros).
- Uma tabela com a quantidade total ingerida de cada nutriente, o respetivo valor objetivo, e a diferença por defeito ou excesso referente a esse valor objetivo.

De realçar que o utilizador também poderá fazer Logout da aplicação através de um ícone, no canto superior direito. De seguida, figura 8, segue-se uma captura de ecrã representativa desta *Dashboard*:

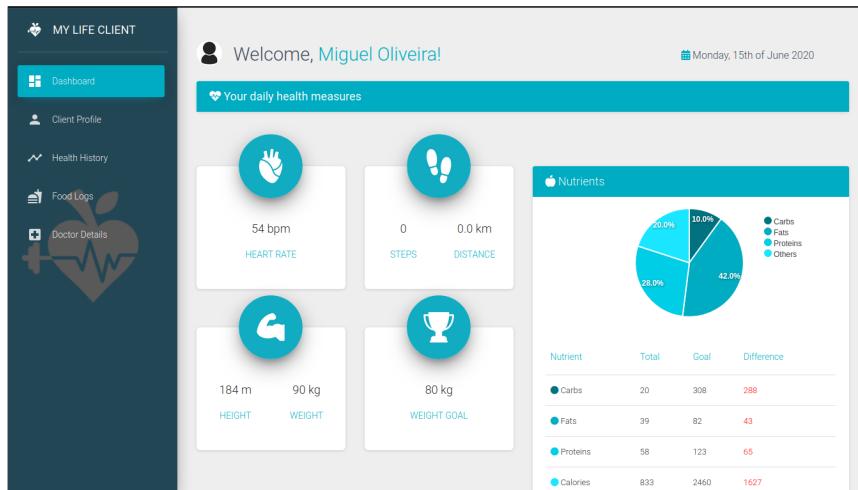


Figura 8: Dashboard do utilizador da aplicação web

### 5.3.3 Perfil

Qualquer utilizador do nosso sistema poderá visualizar diversas informações sobre a sua conta na página de “**Perfil**”, como o email, a altura, o peso, o número de telemóvel, o sexo e se tem alguma condição de saúde relacionada com diabetes ou colesterol alto. Adicionalmente, o utilizador poderá alterar alguns destes parâmetros, bem com a sua palavra-passe, através de um formulário corretamente validado. De seguida, figura 9, segue-se uma captura de ecrã da página de perfil:

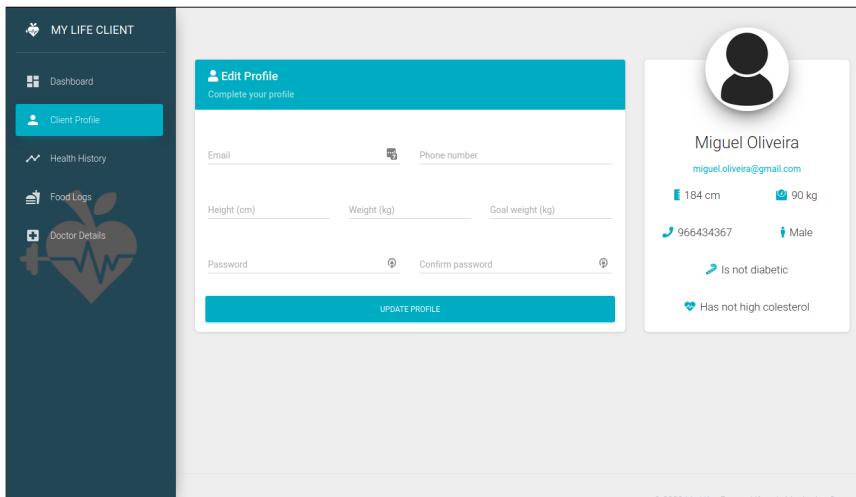


Figura 9: Página de perfil de um utilizador

### 5.3.4 Informação nutricional

Uma das principais funcionalidades que o nosso sistema oferece ao utilizador é a possibilidade de visualização e análise do seu estado geral de saúde. Neste sentido, na página de “**Health History**”, o utilizador tem à sua disposição diversos gráficos e estatísticas, de forma a analisar algumas métricas nutricionais e vitais referentes ao seu estado de saúde geral.

Em primeiro lugar, o utilizador poderá visualizar um gráfico, em estilo de velocímetro, representativo da nossa métrica geral MyLife. O valor desta métrica está inserido em uma de três categorias:

Poor: 0 - 2

Average: 2 - 4

Excellent: 4 - 5

Ainda neste gráfico, o utilizador consegue obter feedback sobre a sua evolução comparativamente à semana anterior.

À direita deste gráfico, está representado um outro gráfico, com características muito semelhantes, mas que representa a qualidade do batimento cardíaco do utilizador. Assim, este valor do batimento cardíaco também está inserido numa de 5 categorias, cujos intervalos de valores variam de indivíduo para indivíduo. Os nomes dessas categorias são os seguintes: Poor, Fair, Average, Good, Excellent. De seguida, figura 10, segue-se uma captura de ecrã ilustrativa destes dois gráficos:

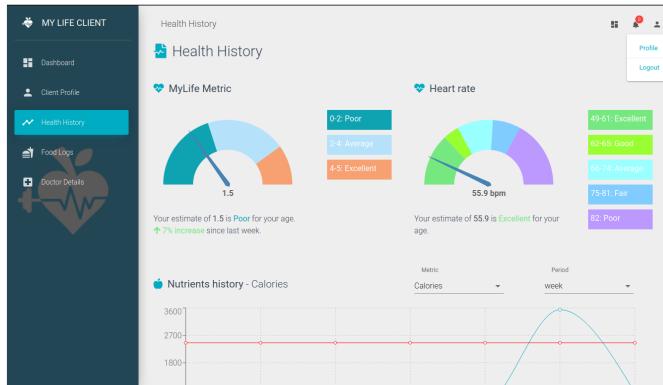


Figura 10: Gráficos representativos da métrica MyLife e do batimento cardíaco

Ainda nesta página, e em baixo destes dois gráficos, estão representados outros dois gráficos, bidimensionais, que representam a evolução, ao longo do tempo, de várias métricas vitais e nutricionais. No primeiro gráfico, o utilizador pode ver a sua evolução referente não só à ingestão de diversos nutrientes, tais como hidratos de carbono, gorduras e proteínas, mas também ao número total de calorias consumidas. No segundo gráfico, está representada a evolução de algumas métricas provenientes da Fitbit, tais como o número de passos percorridos, a distância percorrida, o número de floors e a quantidade de calorias queimadas.

De referir que, em ambos os gráficos, não só está presente uma linha horizontal contínua, a vermelho, referente ao valor objetivo de cada métrica, mas também a possibilidade de visualizar a evolução de cada métrica tendo em conta os valores referentes à última semana, ao último mês, ou aos últimos três meses. De seguida, figura 11, segue-se uma captura de ecrã de ambos os gráficos:



Figura 11: Gráficos com a evolução temporal de diversas métricas vitais e nutricionais

### 5.3.5 Gestão de food logs

Outra funcionalidade bastante interessante da nossa aplicação web é a possibilidade de o utilizador gerir os seus food logs. Essencialmente, esta gestão baseia-se na inserção de novos food logs, ou na visualização de todos os food logs armazenados em persistência.

Por um lado, relativamente à inserção de *food logs*, o utilizador terá de especificar os seguintes parâmetros:

- Data.
- Tipo de refeição (pequeno-almoço, almoço, lanche ou jantar).
- Número de refeições.
- Nome da refeição.

Caso o nome da refeição introduzido não esteja registado no nosso sistema, irá aparecer uma mensagem de erro. Caso contrário, o *food log* será corretamente inserido. De seguida, figura 12, segue-se uma captura de ecrã:

The screenshot shows the 'Food Logs' section of the MY LIFE CLIENT application. On the left is a sidebar with 'Dashboard', 'Client Profile', 'Health History', and 'Food Logs' (which is highlighted in blue). The main area has a teal header bar with the text 'Please, fill in the following parameters:'. Below it, there's a date input field set to '06/15/2020', a radio button group for 'TYPE OF MEAL' (with 'Lunch' selected), a slider for 'SERVINGS' (set to 2), and a text input for 'MEAL' containing 'Hamburger'. A large blue 'ADD FOOD LOG' button is at the bottom right.

Figura 12: Inserção de um novo food log

Por outro lado, e como já referimos, o utilizador poderá visualizar todos os *food logs* inseridos num determinado dia, cada um com as respectivas informações nutricionais. Em cada refeição, poderá visualizar o número total de calorias ingeridas. Se assim o desejar, o utilizador pode remover um ou mais *food logs*. De seguida, figura 13, segue-se uma captura de ecrã desta funcionalidade:

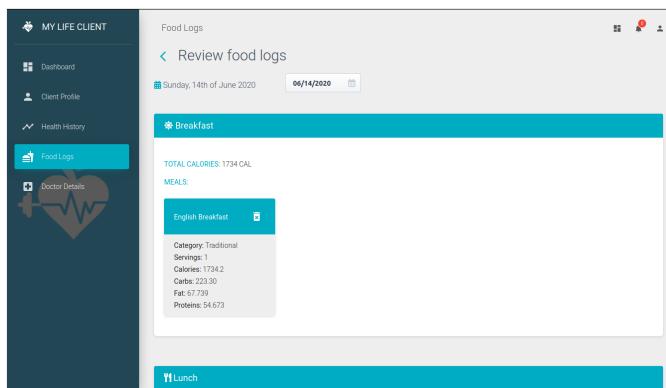


Figura 13: Visualização de food log

### 5.3.6 Médico associado

Em certas situações, um determinado utilizador poderá estar associado a um médico, podendo, deste modo, consultar diversas informações referentes a este médico. Caso pretenda, o utilizador também o poderá remover como seu médico. De seguida, figura 14, segue-se uma captura de ecrã desta funcionalidade:

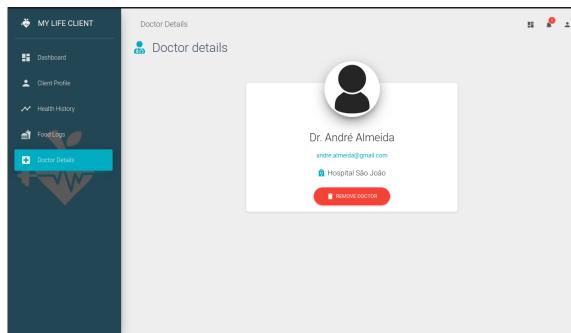


Figura 14: Detalhes do médico

## 5.4 Plataforma Móvel do Cliente/Paciente

A aplicação mobile segue uma estrutura que se baseia em dois componentes principais:

- Food logs e a sua inserção.
- Informação estatística proveniente de:
  - Food logs.
  - Fitbit.

Portanto, para além dos ecrãs e funcionalidades típicas de uma aplicação que está pronta para o mercado (Login, registo, disposição dos dados do user através de perfil e a sua edição), a aplicação também inclui funcionalidades como: **inserção de registos diários de alimentação**, que é impulsionado por algoritmos de Machine Learning para **reconhecimento de comida** e métricas próprias, para melhor situar o cliente na sua jornada de perda/ganho peso. Estes ecrãs e conceitos são, no entanto, melhor descritos em seguida.

### 5.4.1 Perfil

Na secção de Profile, acessível em todos os ecrãs através do ícone no canto superior direito, o cliente consegue observar primeiramente, informação relativa às métricas diárias que conseguimos retirar da Fitbit, como: **passos, batimento cardíaco e distância percorrida**.

O cliente consegue ainda ver a sua fotografia do perfil e o seu nome registado na plataforma.

Para complementar esta página, temos a opção de visualizar um de dois sub-ecrãs:

- Visualização da métrica MyLife.
- Informações detalhadas do perfil.

No caso de o cliente querer visualizar a métrica MyLife (sub-ecrã predefinido), vai-se deparar com uma escala com 3 intervalos: **Poor**, **Average** e **Excellent**, como podemos ver na figura 15. A divisão dos mesmos está delimitada no ecrã, assim como onde o cliente se encontra nessa escala, para mais facilmente perceber se está com um bom índice MyLife (este índice tem em consideração uma combinação de métricas nutricionais e de desporto).



Figura 15: Página de métrica MyLife

Caso o cliente queira visualizar a informação mais detalhada sobre o seu perfil, basta deslizar o ecrã para conseguir aceder a todas as suas informações da conta, assim como verificar o médico que lhe está atribuído e também poder editar as suas informações de perfil, como se pode ver na figura 16.

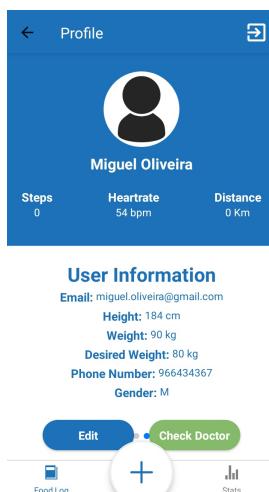


Figura 16: Página de perfil

#### 5.4.2 Estatísticas

Na secção de **Stats**, o cliente consegue observar 2 páginas distintas, como podemos concluir pela figura 17. A primeira página apresenta opções relativamente aos gráficos relacionados com atividade física, cuja fonte principal de informação são as métricas recolhidas a partir de fitness trackers como a fitbit. Por este motivo, se o utilizador não tiver um tracker conectado à nossa aplicação não conseguirá visualizar as opções nesta página.

Nesta página temos a opção de visualizar gráficos relacionados com o ritmo cardíaco, nº de passos, nº de andares, distância percorrida e calorias consumidas. Ao selecionar uma das opções somos redirecionados para uma página onde é apresentado um gráfico com a evolução da métrica ao longo de um intervalo de tempo, sendo possível escolher semana, mês ou 3 meses para o intervalo de tempo (ex. Evolução do nº de passos do cliente ao longo de uma semana). Além disso, também é apresentado numa lista o histórico da última semana relativamente à métrica selecionada.

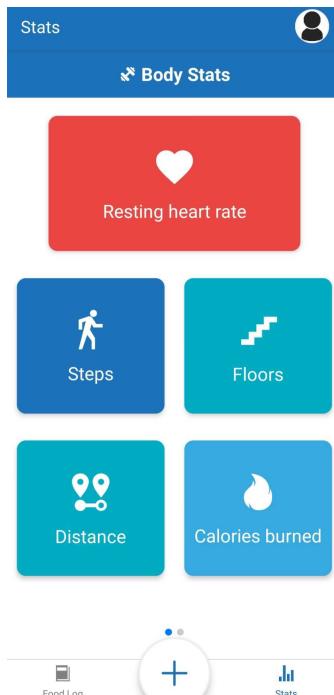


Figura 17: Página de Body Stats, com as várias opções disponíveis

Na opção de ritmo cardíaco, além da informação referida anteriormente, também fornecemos uma escala onde são apresentadas as várias zonas onde se pode inserir: **Excellent, Good, Average, Fair e Poor** fornecendo feedback ao utilizador relativamente ao estado do seu ritmo cardíaco, como podemos observar na figura 18.

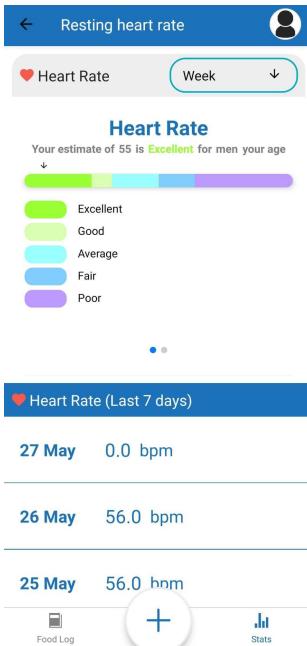


Figura 18: Página de Heart Rate Stats

A segunda página apresenta opções relativamente aos gráficos relacionados com os hábitos alimentares do cliente. Temos a opção de visualizar gráficos relacionados com **Proteínas, Hidratos de carbono, Gorduras e Calorias**. Semelhante à primeira página, ao selecionar uma das opções somos redirecionados para uma página onde é apresentado um gráfico com a evolução da métrica ao longo de um intervalo de tempo e é apresentado numa lista o histórico da última semana relativamente à métrica selecionada.

Através destes gráficos, desenvolvidos com a biblioteca **react native chart kit**, a secção de **Stats** apresenta a evolução do cliente ao longo do tempo em várias métricas.

#### 5.4.3 Visualização de food logs

Tal como foi supracitado, uma das componentes principais da aplicação são os “**food logs**”. Um **food log** é um registo diário das refeições ingeridas num dia.

Na página dos food logs, o cliente encontra inicialmente uma diferença entre as calorias que deve ingerir num dia (calculado a partir do peso que deseja atingir) e as que já ingeriu. Esta diferença tem duas opções de coloração: verde (se ainda faltarem calorias para atingir o objetivo) e vermelho (caso já tenha avançado o objetivo).

Nesta página, o **utilizador** encontra uma divisão em quatro blocos principais de refeições, sendo estes: **Breakfast, Lunch, Dinner e Snacks**, correspondendo a pequeno-almoço, almoço, jantar e lanches intermédios, respetivamente.

Em cada “bloco” conseguimos visualizar as refeições aí consumidas e, conseguimos retirar informação das calorias: por refeição e pelo bloco inteiro. Em cada bloco temos ainda a opção de adicionar comida clicando no botão: “**+ Add Food**”, sendo que vamos ser redirecionados para a página de inserção de food logs mas com o bloco respetivo já escolhido, facilitando então o workflow do cliente.

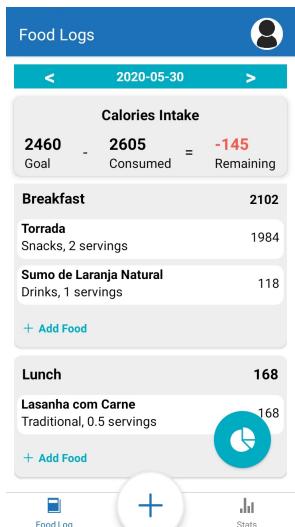


Figura 19: Página de um food log

O utilizador pode ainda alterar o dia que está a visualizar através do uso das setas "<" ou ">", para recuar ou avançar um dia, respetivamente.

Através do botão no canto inferior direito (representativo de um pie chart), acedemos a um ecrã de estatísticas representativas do dia em questão. Este ecrã tem dois sub-ecrãs, sendo que:

- O primeiro corresponde ao progresso nos objetivos diários dos nutrientes (Calorias, hidratos de carbono, gorduras e proteínas) do utilizador.
- O segundo corresponde a um gráfico do tipo “Pie chart”, que indica a percentagem na totalidade de macronutrientes consumidos nesse dia.

#### 5.4.4 Inserção de food logs

A inserção de food logs é uma das funcionalidades mais importantes do nosso sistema, pois permite analisarmos os hábitos alimentares dos nossos clientes e fornecer feedback acerca da evolução dos mesmos. Por este motivo e devido ao feedback dado pelos nossos docentes, esta página sofreu várias mudanças ao longo do ciclo de desenvolvimento com o objetivo de tornar mais intuitiva e mais fácil para os utilizadores inserirem um novo food log.

Através do botão + colocado na navbar em baixo conseguimos aceder à página de inserção através de qualquer secção da aplicação. Após a página carregar, é calculado automaticamente o dia atual, o número de porções e o momento de refeição em que se situa o cliente consoante o tempo atual (ex. Se forem 10h00 a aplicação colocará por defeito **Breakfast** como momento de refeição). Estes valores por defeito permitem ao cliente uma inserção rápida de um novo food log, necessitando apenas de inserir o nome da refeição que ingeriu.

Além do botão +, também é possível aceder a esta página através da página de listagem de food logs. Em cada momento de refeição representado (**Breakfast**, **Lunch**, **Dinner** e **Snack**) existe um botão **Add Food log** que irá redirecionar para a secção de inserção de

um novo food log, tal como foi referido na secção 5.4.3. Ao sermos redirecionados para a página, o momento de refeição e a data serão aqueles que seleccionámos na listagem de food logs.

Figura 20: Página de Inserção de um novo food log

Nesta página, que pode ser visualizada na figura 20, temos várias opções para selecionar a refeição ingerida. Podemos inserir o nome da refeição no campo de input complementado com autocomplete ou selecionar um dos 3 botões apresentados no lado direito deste campo. Ao selecionar o primeiro botão, identificado com uma câmara fotográfica, somos redirecionados para a página **MyLife Food Detector** onde é possível reconhecer refeições através de fotografias. Após selecionar uma imagem o sistema irá identificar a comida presente e as informações nutricionais correspondentes e o cliente poderá adicionar essa refeição à inserção de um novo food log.

Ao selecionar o segundo botão, identificado com um código de barras, somos redirecionados para a página **Barcode Scanner** onde é possível reconhecer alimentos através do seu código de barras. Após o código de barras ser reconhecido, o sistema irá identificar o alimento e as informações nutricionais correspondentes e o cliente poderá adicionar esse alimento à inserção de um novo food log.

Por fim, ao selecionar o terceiro botão, identificado com um +, somos redirecionados para a página **New Meal** onde é possível adicionar uma nova refeição ao sistema. Se o alimento que o cliente ingeriu ainda não existir na nossa base de dados, esta é a opção perfeita, permitindo ao utilizador criar um novo alimento, adicionando os ingredientes que quiser e recebendo feedback em tempo real sobre as informações nutricionais desse alimento.

Ao clicar no botão **Confirm** o food log é adicionado à listagem e o cliente poderá receber feedback sobre esta nova adição (ex: "This food is high on protein") permitindo assim ao utilizador perceber em tempo real se os seus hábitos alimentares estão num bom caminho ou não.

## 5.5 Plataforma do Médico

### 5.5.1 Login e Dashboard

Como vimos na secção 5.3.1, e à semelhança de um utilizador normal, um médico também se poderá autenticar no nosso sistema através da mesma página de Login. No entanto, visto que desempenha um papel diferente no nosso sistema, e através de código dinâmico, irá ser redirecionado para uma interface diferente daquela já apresentada para o utilizador.

Ao autenticar-se corretamente, o médico será redirecionado para a página de Dashboard, presente na seguinte figura , onde poderá navegar para as restantes páginas da sua interface.

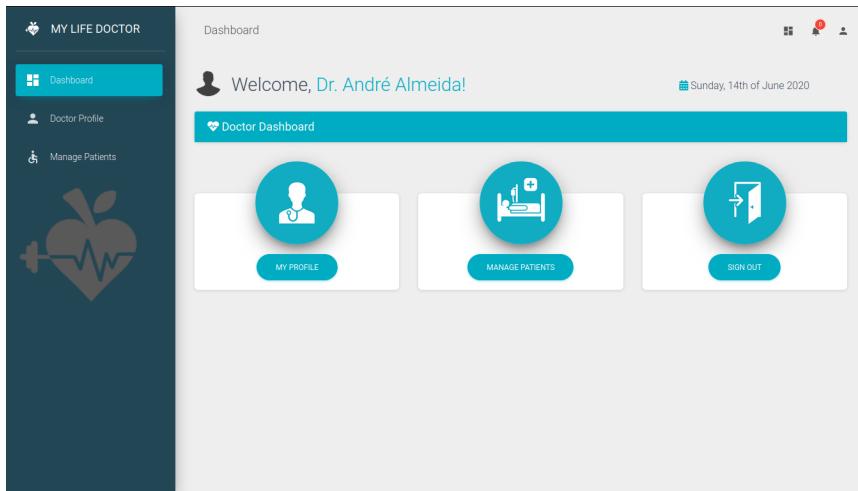


Figura 21: Página de Dashboard de um médico

### 5.5.2 Perfil

À semelhança do utilizador, também o médico poderá consultar e editar diversas informações sobre a sua conta, como o email e a palavra-passe. De seguida, figura 22, segue-se uma captura de ecrã desta página:

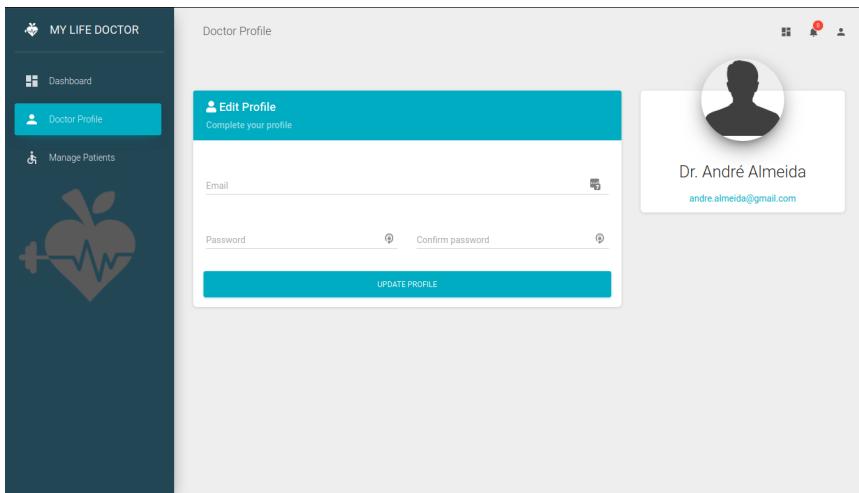


Figura 22: Página de perfil de um médico

### 5.5.3 Gestão de pacientes

A principal funcionalidade disponível a qualquer médico da nossa plataforma é, sem dúvida, a possibilidade de gerir a sua lista de pacientes associados. Neste sentido, quando um médico navega até à página “**Manage Patients**”, será apresentada a lista de todos os seus pacientes associados, cada um com o respetivo nome, email, número de telemóvel e sexo. Adicionalmente, o médico terá à sua disposição, para cada paciente, dois botões, de forma a visualizar os respetivos detalhes ou eliminar esse paciente da sua lista, como se observa na seguinte captura de ecrã, figura 23.

Name	Email	Phone Number	Sex	Details	Remove
Miguel Oliveira	miguel.oliveira@gmail.com	966434367	M		
Vasco Almeida	vasco.almeida@gmail.com	936545567	M		
Albertina Matos	albertina.matos@gmail.com	908000458	F		
Alberto Matos	alberto.matos@gmail.com	910002068	M		
Miguel Pedroso	miguel.pedroso@gmail.com	915005009	M		
Miguel Matos	miguel.matos@gmail.com	930438012	M		

Figura 23: Lista de pacientes associados a um médico

Caso o médico clique no ícone de remover um paciente, irá aparecer a seguinte mensagem de confirmação da ação, figura 24:

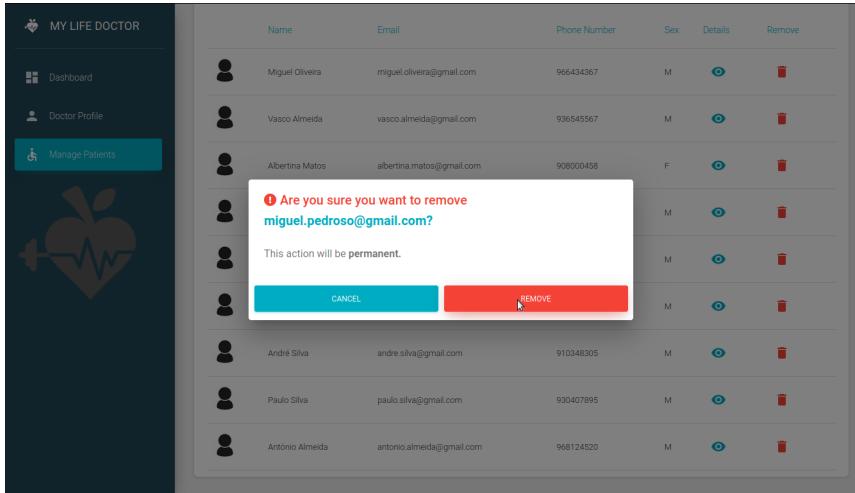


Figura 24: Confirmação da remoção de um paciente

Caso o médico clique no ícone de visualizar os detalhes de um determinado paciente, irá ser redirecionado para uma página com informação nutricional e vital sobre esse indivíduo, muito semelhante à página já apresentada para o utilizador na secção 5.3.4. Como se observa nas seguintes capturas de ecrã, figuras 25 e 26, os gráficos são iguais aos da secção 5.3.4, pelo que não iremos voltar a descrever exaustivamente cada um deles.

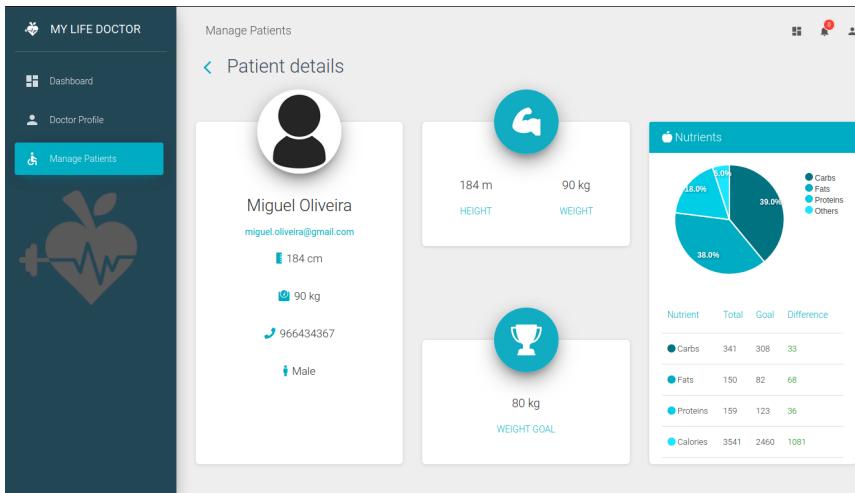


Figura 25: Gráficos representativos da métrica MyLife e do batimento cardíaco de um paciente

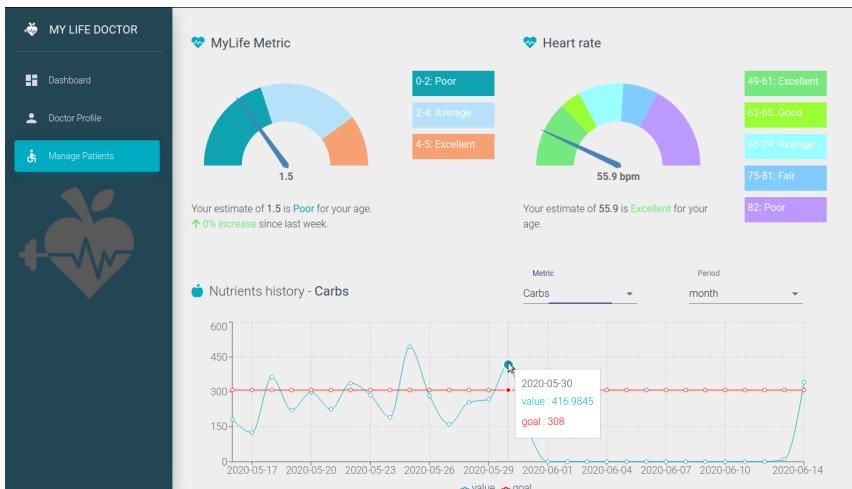


Figura 26: Gráficos com a evolução temporal de diversas métricas vitais e nutricionais de um paciente

Esta funcionalidade é bastante interessante, uma vez que permite ao médico visualizar o estado geral de saúde de um determinado paciente, bem como a respetiva evolução ao longo do tempo.

O médico pode ainda adicionar novos pacientes à sua lista de pacientes associados. Para tal, apenas tem de clicar no botão “**Add new patient**”, e colocar o endereço de email de um utilizador registado no sistema. Caso esse utilizador já tenha um outro médico associado, a operação não será realizada. De seguida, figura 27, segue-se uma captura de ecrã ilustrativa desta operação:

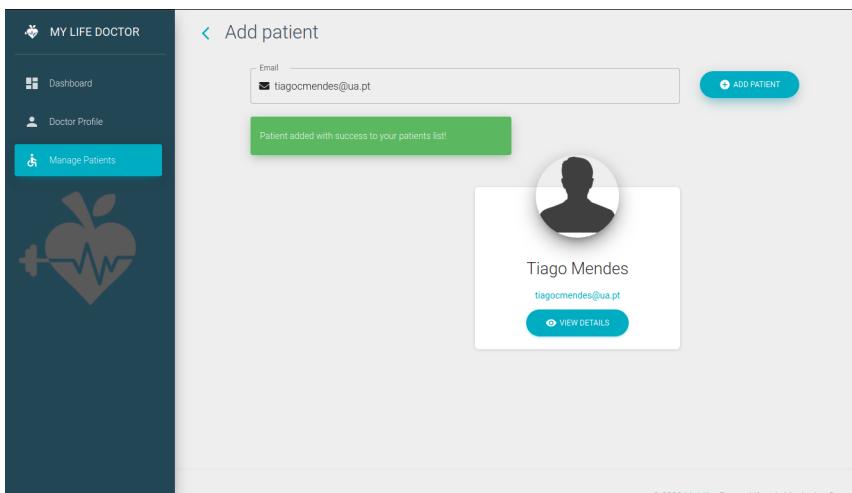


Figura 27: Adição de um novo paciente à lista de pacientes associados

## 5.6 Plataforma do Administrador

### 5.6.1 Login e Dashboard

Como vimos nas secções 5.3.1 e 5.5.1, também um administrador do hospital se poderá autenticar no nosso sistema através da mesma página de Login. No entanto, visto que desempenha um papel diferente no nosso sistema, e através de código dinâmico, irá ser redirecionado para uma interface diferente daquela já apresentada para o utilizador e para o médico.

Ao autenticar-se corretamente, o administrador do hospital será redirecionado para a página de Dashboard, presente na seguinte figura 28, onde poderá navegar para as restantes páginas da sua interface.

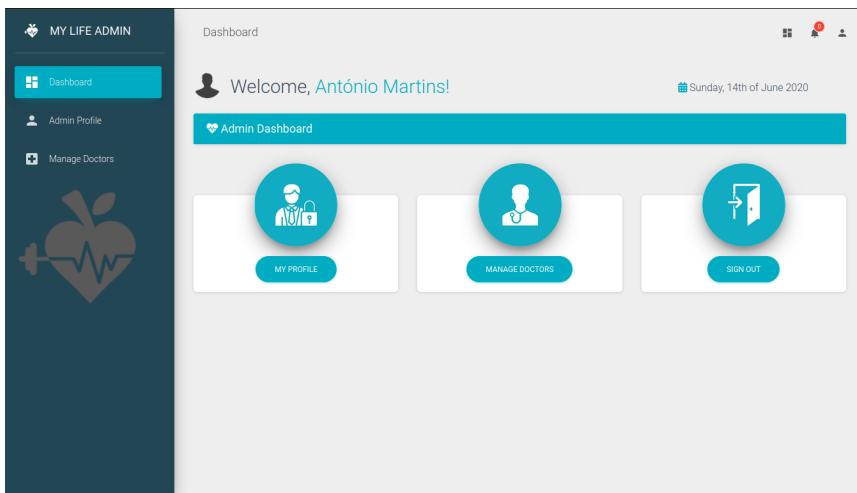


Figura 28: Dashboard do administrador

### 5.6.2 Perfil do administrador

Muito semelhante à página de perfil do médico, também o administrador do hospital poderá consultar e editar diversas informações sobre a sua conta, como o email e a palavra-passe. De seguida, figura 29 segue-se uma captura de ecrã desta página de perfil:

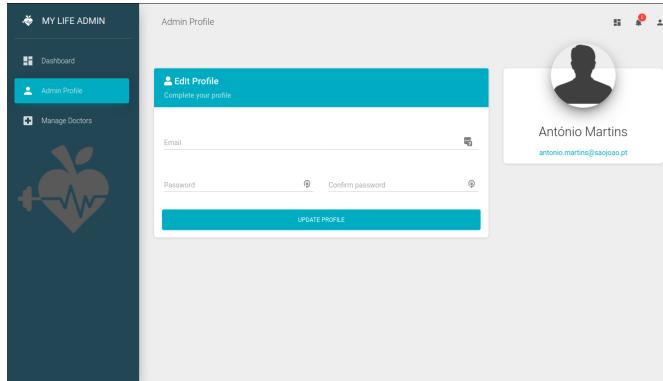


Figura 29: Página de perfil do administrador do hospital

### 5.6.3 Gestão de médicos

A principal funcionalidade disponível a qualquer administrador de um hospital na nossa plataforma é, sem dúvida, a possibilidade de gerir a lista de médicos associados ao seu hospital. Neste sentido, quando um médico navega até à página “**Manage Doctors**”, será apresentada a lista de todos os médicos associados ao seu hospital, cada um com o respetivo nome, email e nome do hospital. Também irá ter à sua disposição, para cada médico associado, uma opção de o remover do hospital. De seguida, figura 30, segue-se uma captura de ecrã da lista de médicos.

Name	Email	Hospital	Remove
André Almeida	andre.almeida@gmail.com	Hospital São João	

Figura 30: Lista de médicos associados a um hospital

Para além disto, o administrador do hospital também poderá adicionar um novo médico ao sistema, através do preenchimento de um formulário corretamente validado. Neste sentido, para registar um novo médico, terá de especificar os seguintes parâmetros:

- Primeiro e último nome.

- Email.
- Número de telemóvel.
- Data de nascimento.
- Palavra-passe.
- Confirmação da palavra-passe

De seguida, figura 31, segue-se uma captura de ecrã com o formulário de registo de um novo médico que, após todos os campos terem sido validados, irá ser adicionado à lista de médicos registados no hospital.

The screenshot shows the 'MY LIFE ADMIN' application interface. On the left is a sidebar with 'Dashboard', 'Admin Profile', and 'Manage Doctors' (which is highlighted in blue). The main area is titled 'Manage Doctors' and has a sub-section 'Add doctor'. It contains fields for 'First name \*' and 'Last name \*', both with a magnifying glass icon. Below that are fields for 'Email' and 'Phone number \*', followed by a date input field showing '06/14/2020'. At the bottom is a large blue button labeled '+ ADD DOCTOR'.

*Figura 31: Formulário de registo de um novo médico*

## 6 Discussão de Resultados

Desde o início deste projeto tivemos três objetivos em mente que gostávamos que se verificassem na solução final:

- A solução devia ser intuitiva e simples
- A solução devia ter uma performance elevada
- A solução devia ser útil para o utilizador

De um modo geral, atingimos os nossos objetivos e conseguimos apresentar um projeto que vai de encontro aos tópicos referidos anteriormente, onde é possível monitorizar hábitos de alimentação e de exercício físico, promovendo assim um estilo de vida saudável.

Com a ajuda dos docentes e do nosso orientador, conseguimos identificar alguns problemas de usabilidade ao longo do desenvolvimento do projeto, especialmente na funcionalidade de inserção de um novo food log, diminuindo o número de cliques necessários para a submissão de um novo food log.

Uma das funcionalidades que mais se destaca no nosso projeto é a integração com algoritmos de *Machine Learning*, o que permitiu de uma forma rápida e fiável reconhecer alimentos na nossa aplicação. Uma das maiores dificuldades no reconhecimento de alimentos através de uma imagem é calcular a quantidade presente desse alimento.

De forma a apresentar uma solução a este problema, após selecionar uma fotografia da refeição e esta ser reconhecida pelo sistema, o utilizador pode selecionar o número de porções que ingeriu da refeição, resultando numa aproximação mais realista à quantidade ingerida pelo utilizador.

Tal como referimos na secção 2.1, uma das aplicações líder de mercado nesta área e que mais facilmente se compara ao MyLife é o MyFitnessPal. Assim, estabelecendo uma comparação entre as duas aplicações, na nossa opinião, o MyLife é superior em simplicidade de uso e na deteção de comida de forma rápida, pois apesar do MyFitnessPal permitir reconhecer alimentos por código de barras, não permite reconhecê-los através de uma imagem. Por outro lado, o MyFitnessPal é uma aplicação mais robusta, por se encontrar há vários anos no mercado e apresentar funcionalidades adicionais às nossas como receitas personalizadas, diversos gráficos de visualização de progressão, entre outras.

Outra aplicação que se destaca, neste caso na área de monitorização de métricas de saúde é o Google Fit. Na nossa opinião a nossa aplicação oferece uma solução completa nesta área, permitindo conectar com fitness trackers externos e apresentar informação relativa à evolução de métricas de saúde como passos, distância percorrida e batimento cardíaco. Contudo, aplicações como o Google Fit, permitem uma visão mais detalhada das métricas que referimos anteriormente e mesmo de outras métricas que não estão presentes na nossa aplicação.

Apesar de tudo, na nossa opinião, desenvolvemos uma solução bastante completa, num curto espaço de tempo, que apresenta funcionalidades presentes em outras aplicações, mas também funcionalidades inovadoras, indo de encontro aos objetivos definidos inicialmente para este projeto. A nossa estimativa no reconhecimento de comida através de uma imagem acabou por ser muito realista, resultando numa aproximação correta na maior parte das situações e é superior à alternativa de ter que inserir manualmente o nome da refeição que estamos a ingerir.

É possível consultar o nosso projeto no nosso website público (<https://my-life-ua.gitlab.io>) e o vídeo promocional do mesmo, no seguinte url: <https://www.youtube.com/watch?v=kf0Zt1jCs4U>.

## 7 Gestão do Projeto

Desde o início do desenvolvimento do projeto, decidimos adotar medidas e um fluxo de trabalho o mais profissional possível, de forma a assegurar um desenvolvimento mais rápido mas que, ao mesmo tempo, providenciasse um produto mais estável e de qualidade assegurada.

Para isso, adotámos várias medidas de *Continuous Integration (CI)* e *Continuous Deployment (CD)*.

A implementação de um fluxo de *CI* consistiu no desenho e na escrita de vários testes aos vários níveis da *stack de software*, de mecanismos de formatação e lint automático, terminando com compilação e empacotamento de cada módulo. A execução destes passos para cada incremento de código no repositório permitiu-nos visualizar constantemente o progresso, bem como garantir a manutenção da qualidade e funcionalidade da solução.

Por outro lado, o processo de *CD* permitia colocar, em poucos minutos, a solução atual no servidor ao qual tínhamos acesso, de forma a que pudéssemos testar e visualizar as alterações.

Desta maneira, recorrendo à plataforma *Gitlab*, conseguimos obter e utilizar um serviço central onde concentrarmos o código, as operações de integração e *deployment*, assim como análise e discussão, em equipa, dos mesmos. Podemos visualizar as mesmas nas figuras 32 e 33:

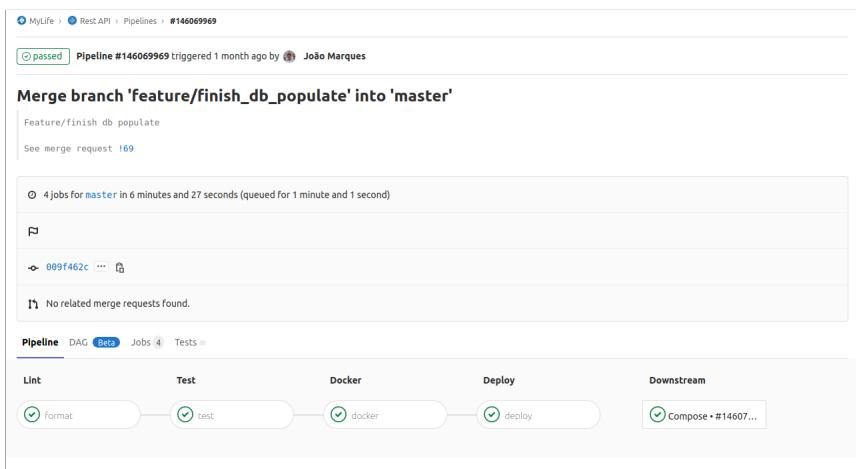


Figura 32: Exemplo de pipeline de integração e deployment

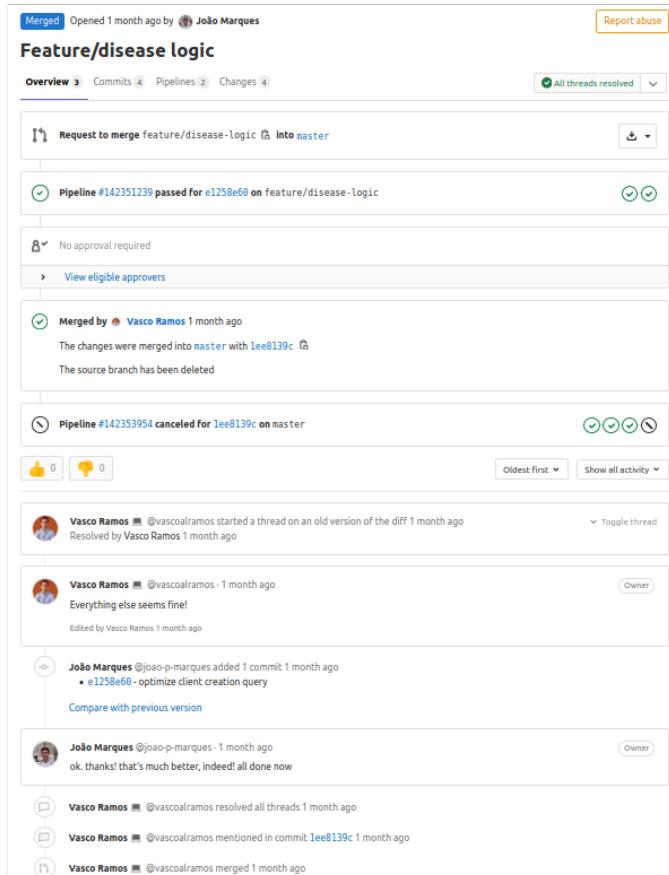


Figura 33: Exemplo de discussão de código em equipa

Por fim, optámos por isolar cada módulo num *container*, cuja configuração era partilhada entre toda a equipa, e era o meio pelo qual o serviço era executado tanto em **desenvolvimento** como em **produção**. Este isolamento, bem como a facilidade na partilha entre as configurações dos mesmos, permitiu-nos assegurar a uniformidade entre os ambientes de desenvolvimento, não só entre os vários elementos da equipa, mas também entre os ambientes de desenvolvimento e produção. Esta estabilidade de ambientes, bem como a segurança e as vantagens a nível de replicação que proporciona a utilização de *containers* permitiu-nos satisfazer mais rapidamente os requisitos estabelecidos para o projeto.

## 8 Conclusões e Trabalho Futuro

O principal objetivo deste projeto foi sempre o de desenvolver um sistema completo, eficiente, robusto e realista, capaz de ser uma mais valia para todas as pessoas que pretendam, de alguma forma, melhorar o seu estilo de vida pessoal, tornando-se mais ativas e saudáveis. Como disse o nosso orientador, Prof. Dr. Carlos Costa, nós somos o que comemos, o que bebemos, o ar que respiramos e o exercício que praticamos, pelo que o sistema MyLife pretende ajudar o consumidor final a maximizar todas estas variáveis.

De um modo geral, estamos convictos de que desenvolvemos um projeto completo, estando muito próximo de uma versão pronta a ser comercializada e consumida por diferentes pessoas, nos mais variados contextos. Apesar de, neste momento, apenas estar num estado de prova de conceito, o nosso sistema integra funcionalidades interessantes, sendo também apresentado de uma forma simples e intuitiva aos utilizadores.

O desenvolvimento deste projeto foi, sem dúvida, uma mais valia para todos os elementos do grupo, no sentido em que nos permitiu não só desenvolver novas competências, mas também aprofundar conhecimentos adquiridos ao longo deste primeiro ciclo de estudos em Engenharia Informática. Através da aplicação de dinâmicas e metodologias ágeis de desenvolvimento de *software*, fomos capazes de melhorar tanto as nossas competências tecnológicas, como também as nossas competências interpessoais para com os restantes elementos do grupo. Sem dúvida, este projeto deu-nos bases importantíssimas para os nossos futuros académico, profissional, social e pessoal, pelo que todos os elementos do grupo valorizam a sua importância no âmbito desta licenciatura.

Para terminar, gostaríamos de referir algum trabalho futuro que consideramos que seria interessante desenvolver:

- Em primeiro lugar, tendo em conta as limitações do atual módulo de reconhecimento de comida, seria interessante melhorar o desempenho deste em situações de reconhecimento de vários alimentos numa fotografia, permitindo fornecer os respetivos valores nutricionais de uma forma mais precisa e realista.
- Em segundo lugar, aumentar o conjunto de dados das nossas refeições, tornando-o mais variado. Desta forma, conseguiríamos ter um maior universo de informação e prestar um melhor serviço aquando da identificação de refeições, seja por pesquisa manual, código de barras ou mesmo imagem.
- E, por fim, após diálogo com o nosso orientador, seria vantajoso a existência de um mecanismo que permitisse a qualquer utilizador partilhar a sua informação nutricional com qualquer amigo, familiar ou mesmo profissional de saúde, através de um link. Desta forma, seria possível remover do nosso sistema a lógica de negócio associada à conta do médico, o que diminui a complexidade intrínseca ao mesmo e aumenta a segurança e proteção de dados do utilizador, uma vez que só quem tivesse esse link de partilha é que poderia aceder à informação relativa ao utilizador.



## Referências

- [1] D. J. Attokaren, I. G. Fernandes, A. Sriram, Y. V. S. Murthy, and S. G. Koolagudi, "Food classification from images using convolutional neural networks," in *TENCON 2017-2017 IEEE Region 10 Conference*, pp. 2801-2806, 2017.
- [2] P. Rodriguez, "Food Classification with Deep Learning in Keras / Tensorflow." <https://github.com/stratospark/food-101-keras>. Accessed: 2020-06-20.
- [3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Real-time multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [4] S. Chen and R. Yang, "Pose trainer: Correcting exercise posture using pose estimation," *GitHub Repository*. Available online: <https://github.com/stevenzchen/pose-trainer> (Accessed: 2020-06-20), 2018.
- [5] "YOLO: Real-Time Object Detection." <https://pjreddie.com/darknet/yolo>. Accessed: 2020-06-19.
- [6] B. Cheung, "YOLO for Real-Time Food Detection." <https://bennycheung.github.io/yolo-for-real-time-food-detection>. Accessed: 2020-06-19.
- [7] "Fitbit - Web API Reference." <https://dev.fitbit.com/build/reference/web-api>. Accessed: 2020-06-19.
- [8] "Darknet: Open Source Neural Networks in C." <https://pjreddie.com/darknet>. Accessed: 2020-06-19.
- [9] T. H. Trieu, "Darkflow," *GitHub Repository*. Available online: <https://github.com/thtrieu/darkflow> (Accessed: 2020-06-19), 2018.
- [10] Open Food Facts contributors, "Open Food Facts Dataset." <https://world.openfoodfacts.org/data>. Accessed: 2020-06-11.
- [11] Open Food Facts contributors, "Open Food Facts REST API." <https://wiki.openfoodfacts.org/API>. Accessed: 2020-06-11.