



Automatização do processo de criação de planos de treino

ANTÓNIO MANUEL MARQUES FERREIRA

Outubro de 2019

Automatização do processo de criação de planos de treino

António Ferreira

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Computer Systems**

Orientador: Dr. Jorge Coelho

Coorientador: Davide Teixeira

Júri:

Presidente:

Ricardo Severino, Doutorado, ISEP

Vogais:

Paulo Baltarejo Sousa, Doutorado, ISEP

Jorge Coelho, Doutorado, ISEP

Davide Teixeira, Licenciado, Universidade Portucalense

Porto, 13 de Outubro de 2019

Página em branco [apagar este comentário]

Dedicatória

Dedicado à minha família, é por eles e graças a eles que aqui estou.

Página em branco [apagar este comentário]

Resumo

O mercado *fitness* e o exercício físico têm vindo a crescer ao longo dos anos, não só em Portugal, mas mundialmente. Atualmente, há uma maior preocupação pela boa forma física e pelo bem-estar, resultando num grande investimento neste mercado. Este investimento levou a um aumento do número de ginásios, marcas de roupa de desporto e de alimentos saudáveis e, no setor tecnológico, à criação de aplicações móveis de suporte ao exercício físico.

Complementarmente, surgiu o conceito de *wearable*, com os *smartwatches* e as *fitness bands*, que permitiam obter dados sobre um dado exercício como, por exemplo, a distância, as calorias perdidas, o tempo e mesmo os batimentos cardíacos.

Aparentemente, com o crescente número de aplicações *fitness* gratuitas e com a quantidade de informação disponível na internet, o papel do assistente pessoal de treino, vulgarmente conhecido como Personal Trainer (PT), tende a tornar-se cada vez menos relevante.

No entanto, o trabalho do PT pode ser complementado com estas tecnologias no sentido de não eliminar o seu papel, mas de o tornar mais eficaz.

Após uma pesquisa exaustiva de aplicações para complementar o trabalho do PT, constatou-se que existem muito poucas opções disponíveis.

Perante o cenário descrito, esta dissertação centra-se na criação/evolução de uma aplicação móvel, onde o PT poderá centralizar toda a informação dos seus clientes, tendo a possibilidade de automatizar todas as fases do seu trabalho, por exemplo, através da criação automática de treinos baseada na intensidade, zona do corpo e frequência cardíaca. A aplicação conciliada com *wearables* destinados a perceber o estado e a performance do utilizador dará informações que permitirão ao PT um trabalho mais personalizado e eficaz.

Palavras-chave: Personal Trainer, Apple Watch, iOS, aplicação móvel, criação automática de planos de treino

Página em branco [apagar este comentário]

Abstract

Currently there is a greater concern for *fitness* and well-being and as a result there has been a great investment in this market. This investment has led to an increase in the number of gyms, sports clothing brands, food brands directed to healthy foods and in the technological sector to the creation of mobile applications to support physical exercise.

Complementarily, the concept of *wearable* was born, with the *smartwatches* and *fitness bands* that allowed to obtain data about a given exercise, as for example: distance, calories burned, time, or even heart beats.

With the growing number of *fitness* applications available for free, and with the amount of information available on the internet, the role of Personal Trainer (PT) tends to become less and less relevant.

However, the work of the PT can be complemented with these technologies making his work more effective.

After carefully reviewing the mobile application market, we found that the PT has few options to help him maximize his work.

Therefore, we propose the creation/evolution of a mobile application, in which the PT can centralize all the information of his clients, with the possibility to automate phases of his work, as the automatic creation of training based on intensity, body area and ideal heart rate zone. *Wearables* are used to provide customer specific information and make the PT's work more effective.

Keywords: Personal Trainer, Apple Watch, iOS, mobile application, automatic creation of workouts

Página em branco [apagar este comentário]

Agradecimentos

Gostaria de agradecer a todos que contribuíram diretamente e indiretamente para a concretização deste projeto, amigos, colegas de trabalho, orientadores e principalmente à minha família que me proporcionou a possibilidade de finalizar o mestrado e este projeto.

O vosso apoio foi incondicional, nos bons e maus momentos. Numa jornada que nunca foi fácil, mas que se tornou possível com bastante trabalho e força de vontade, algumas dores de cabeça e noites mal dormidas, mas no final um prazer imenso por esta meta alcançada.

Página em branco [apagar este comentário]

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema	1
1.3	Objetivos	2
1.4	Resultados Esperados	3
1.5	Análise de Valor	3
1.6	Estrutura	3
1.7	Publicação	4
1.8	Sumário	4
2	Contexto	6
2.1	Contexto e Problema	6
2.1.1	Contexto	6
2.1.2	Problema	7
2.1.3	Restrições Existentes	8
3	Análise de Valor	9
3.1	Introdução	9
3.2	The new concept development model (NCD)	10
3.2.1	Parte interior do modelo (Inner Part)	11
3.2.2	Parte exterior do modelo (External environmental factors)	13
3.3	Valor, Valor Percecionado e Valor para o Cliente	14
3.3.1	Valor	14
3.3.2	Valor Percecionado	14
3.3.3	Valor para o Cliente	15
3.4	Benefícios e Sacrifícios para as Partes Interessadas	15
3.5	Proposta de Valor	15
3.6	Modelo Canvas	16
3.6.1	Mercado Alvo (Customer Segments)	17
3.6.2	Proposta de Valor (Value Propositions)	17
3.6.3	Canais (Channels)	17
3.6.4	Relações com os clientes (Customer Relationships)	18
3.6.5	Perspetivas de Receita (Revenue Streams)	18
3.6.6	Recursos Chave (Key Resources)	18
3.6.7	Atividades Chave (Key Activities)	18
3.6.8	Perspetiva de Custos (Cost Structure)	19
3.6.9	Parcerias chave (Key Partners)	19
3.7	Rede de Valor	19
3.8	Modelo AHP	20

4	Estado de Arte	21
4.1	Smartwatches	21
4.1.1	Apple Watch	22
4.1.2	Samsung Gear S3	22
4.1.3	TAG Heuer Connected Modular	23
4.2	Fitness Bands	23
4.2.1	Xiaomi Mi Band 4	24
4.2.2	Fibit Charge HR	24
4.2.3	Samsung Gear Fit 2	25
4.2.4	Sony SmartBand	25
4.3	Aplicações de Auxílio ao exercício físico (Fitness)	26
4.3.1	Zova	26
4.3.2	Nike + Training Club	26
4.3.3	Runkeeper	27
4.4	Comparações finais	28
4.4.1	Smartwatches	28
4.4.2	Fitness bands	29
4.4.3	Aplicações de Auxílio ao exercício físico	30
5	Análise	33
5.1	Design da solução	33
5.1.1	Requisitos Funcionais	36
5.1.2	Requisitos Não Funcionais	37
5.2	Design Arquitetural	38
5.2.1	Model View Controller (MVC)	38
5.2.2	Model View ViewModel (MVVM)	39
5.3	Fluxograma.....	40
5.4	Design da Interface Gráfica.....	41
5.5	Criação de planos adaptados e dinâmicos	48
5.5.1	Exercícios do Treino	48
5.5.2	Zonas de batimentos cardíacos	50
5.5.3	Objetivos e Zonas de batimentos cardíacos	51
5.5.4	HIIT.....	52
5.5.5	Algoritmo de Geração de Treino	52
5.6	Soluções/Abordagens Existentes	53
5.7	Avaliação das Soluções/Abordagens existentes	54
5.8	Programas necessários	56
5.8.1	Xcode	56
5.9	Equipamentos necessários	59
5.9.1	iPhone	59
5.9.2	Apple Watch	60
5.10	Tecnologias	62
5.10.1	Objective-C	62
5.10.2	HealthKit	66

5.10.3	Watch Connectivity	67
6	Implementação	70
6.1	HealthKit	70
6.2	Watch Connectivity	77
6.2.1	Ciclo de vida de uma aplicação para Apple Watch	83
6.3	Algoritmo de Geração de Treino (Parte técnica)	85
7	Conclusões.....	91
8	Referências.....	94

**Inserir página em branco apenas se necessário de modo
a que a próxima secção comece numa página à direita**

Lista de Figuras

Figura 1 - Processo de Inovação.....	9
Figura 2 - New Concept Development Model.....	10
Figura 3 - Modelo Canvas.....	16
Figura 4 - Modelo de Domínio	34
Figura 5 - Diagrama de Implantação	35
Figura 6 - Diagrama referente à interligação do Apple Watch e do iPhone (Gusgård, 14)	35
Figura 7 - Diagrama de Componentes.....	36
Figura 8 - Diagrama de Casos de Uso.....	37
Figura 9 - Padrão MVC	39
Figura 10 - Padrão MVVM.....	40
Figura 11 - Fluxograma da aplicação.....	41
Figura 12 - Primeiro ecrã da aplicação.....	42
Figura 13 - Ecrã do Workout selecionado	43
Figura 14 - Ecrã de criação de um Workout.....	44
Figura 15 - Ecrã da lista de exercícios do Workout	45
Figura 16 - Ecrã de criação de exercícios para o Workout.....	46
Figura 17 - Ecrã de criação do Workout preenchido	47
Figura 18 - Ecrã Principal da aplicação Apple Watch	48
Figura 19 - Estrutura do Xcode.....	57
Figura 20 - Simulador de Apple Watch Series 2	58
Figura 21 - Simulador de iPhone 7	58
Figura 22 - Vendas de iPhone desde o lançamento do primeiro modelo.....	59
Figura 23 - Versões do iOS mais utilizadas.....	60
Figura 24 - Apple Watch modelo simples	61
Figura 25 - Aplicação Apple Watch	61
Figura 26 - Exemplo de criação de um objeto de classe	63
Figura 27 - Parte pública da criação de uma classe	64
Figura 28 - Parte privada de uma classe	64
Figura 29 - Estrutura de a declaração de métodos	65
Figura 30 - Estrutura de chamada de um método	65
Figura 31 - Página principal da aplicação Saúde	66
Figura 32 - Exemplo de uma Activity Ring.....	67
Figura 33 - Como importar o HealthKit.....	70
Figura 34 - Propriedades privadas do WorkoutManager	70
Figura 35 - Como o Delegate funciona.....	71
Figura 36 – Método que inicializa propriedades/variáveis.....	72
Figura 37 - Método que verifica o estado do HealthKit e a autorização de acesso aos dados..	72
Figura 38 - Exemplo do ecrã de autorização de acesso à informação de saúde	73
Figura 39 - Método de inicialização do HKWorkoutSession	74
Figura 40 - Métodos delegate do HKWorkoutSession	75

Figura 41 - Método que informa o HealthKit da consulta a ser feita para obter dados	76
Figura 42 - Criação de uma consulta dando uma data de início e tipo de dados	76
Figura 43 - Conclusão do treino	77
Figura 44 - Importação do Watch Connectivity	78
Figura 45 - Propriedades privadas da classe Communication Manager	78
Figura 46 - Métodos de inicialização e verificação do estado da ligação entre os dispositivos	79
Figura 47 - Métodos de envio de mensagens	80
Figura 48 - Chaves de identificação de mensagens.....	80
Figura 49 - Métodos do CommunicationManagerDelegate.....	81
Figura 50 - Métodos delegate do WCSessionDelegate	82
Figura 51 - Estado da sessão quando vários dispositivos estão emparelhados com o mesmo iPhone.....	83
Figura 52 - Ciclo de vida de uma aplicação para Apple Watch	84
Figura 53 - Permitir modo Background no Apple Watch	85
Figura 54 - Diagrama de Classes.....	86
Figura 55 - Informação enviada para Apple Watch ao iniciar treino	86
Figura 56 - Temporizador responsável por chamar o método "everySecondAction"	87
Figura 57 - Método everySecondAction.....	87
Figura 58 - Verificação do Batimento cardíaco	88
Figura 59 - Método de aumento de Intensidade do treino	89
Figura 60 - Método de Calculo do batimento cardíaco máximo.....	89
Figura 61 - Cálculo no novo mínimo e máximo cardíaco	90

**Inserir página em branco apenas se necessário de modo
a que a próxima secção comece numa página à direita**

Lista de Tabelas

Tabela 1 - Tabela de Benefícios/Sacrifícios	15
Tabela 2 - Comparação entre <i>Smartwatches</i>	28
Tabela 3 - Comparação entre <i>Fitness bands</i>	29
Tabela 4 - Comparação entre aplicações móveis de auxílio ao treino	30
Tabela 5 - Exemplo de uma lista de Exercícios e suas variáveis	49
Tabela 6 - Tabela de zonas de batimentos cardíacos	50
Tabela 7 - Representação dos Objetivos em relação com as zonas cardíacas	51
Tabela 8 - Exemplo de Questionário sobre a Usabilidade	54
Tabela 9 - Exemplo de <i>Rating</i> da Aplicação móvel	56
Tabela 10 - Metodologias de envio de informação entre os dispositivos	68

Inserir página em branco apenas se necessário de modo a que a próxima secção comece numa página à direita

Acrónimos e Símbolos

Lista de Acrónimos

AHP	Analytic Hierarchy Process
AMOLED	Matriz Ativa de Díodo Orgânico Emissor de Luz
API	Application Programming Interface
ECG	Eletrocardiograma
FFE	Fuzzy Front End
GPS	Sistema de posicionamento global
HIIT	High Intensity Interval Training
HTTPS	Hyper Text Transfer Protocol Secure
iOS	Sistema operativo da Apple para iPhone, iPad e iPod
JSON	JavaScript Object Notation
macOS	Sistema operativo da Apple para todos os computadores Mac
MVC	Model View Controller
MVVM	Model View ViewModel
NCD	New Concept Development
OS	Operative System (Sistema operativo)
PEST	Política, Económica, Social e Tecnológica
PT	Personal Trainer
SDK	Software development kit
tvOS	Sistema operativo da Apple para o Apple TV
UC	<i>Use Case</i> ou Caso de Uso
watchOS	Sistema operativo da Apple para o Apple Watch

**Inserir página em branco apenas se necessário de modo
a que a próxima secção comece numa página à direita**

1 Introdução

O primeiro capítulo servirá para enquadrar o leitor sobre a temática desta dissertação. Para tal será feito um resumo do contexto e do problema abordado, dos objetivos deste projeto e dos resultados esperados.

1.1 Contexto

Ao longo dos anos, a importância do exercício físico e do bem-estar pessoal tem crescido, surgindo mais ginásios e notando-se a tendência de um mercado crescente, (AGAP, 2016). Consequentemente, há um ambiente propício a um maior investimento por parte dos ginásios em maquinaria, espaço e, sobretudo, em PT's, que pretendem obter uma maior quota dos clientes.

Este crescimento é também visível nas marcas, que aproveitam para obter mais valor. Um outro setor que tem crescido exponencialmente é o das aplicações móveis, que tem criado várias oportunidades para o mercado do *fitness* e exercício físico com aplicações de auxílio ao treino, tiram proveito do crescente uso de *Smartphones*. Este crescimento transborda também para a criação de novos dispositivos para o auxílio no treino como os *Smartwatches* e as *Fitness bands*, que surgiram aproveitando o crescimento do mercado.

1.2 Problema

Como resultado da evolução tecnológica, o papel do PT tem vindo a ser parcialmente ameaçado. Cada dia surgem mais aplicações gratuitas de *fitness* que ajudam num treino autónomo, permitindo focar num objetivo específico, muito na linha daquilo que um PT faz. As aplicações exemplificam a execução técnica do/s exercício/s e calculam o seu impacto e nível de adequação ao objetivo definido pelo utilizador.

Como consequência, muitos utilizadores não consideram a compra dos serviços de um PT, uma vez que as aplicações são, maioritariamente, gratuitas e estão disponíveis em qualquer *Smartphone*, seja Android ou iOS. É de notar que uma aplicação não substitui o acompanhamento de um PT com formação, tendo em conta que não é capaz de perceber, por exemplo, a performance técnica do utilizador. No entanto, com a informação disponível na internet, um utilizador pode ser levado a crer que a prática acompanhada do exercício físico não lhe trará grande benefício.

Para reforçar este problema, o PT, normalmente, não utiliza este tipo de tecnologias a seu favor. Com as inúmeras tecnologias disponíveis, como é o caso das aplicações móveis, juntamente com os *wearables*, o PT poderia tornar-se mais eficiente, eficaz e atrativo para os seus clientes (Laurinavicius, 2016).

1.3 Objetivos

O objetivo/meta deste projeto é reforçar a posição do PT e desenvolver uma aplicação móvel direcionada ao mesmo, de forma otimizar o seu trabalho. Ao mesmo tempo, pretende-se que a aplicação seja de fácil compreensão e utilização, uma vez que o utilizador pode não estar familiarizado com as tecnologias.

Para otimizar a atividade do PT, serão estudadas e testadas funcionalidades que poderão, no futuro, ser adicionadas à aplicação móvel iOS **PTrainer** (Apple, PTrainer, 2017) com o objetivo de auxiliar, automatizar e criar um maior dinamismo entre o PT e os seus clientes. Isto dará mais informação ao PT, uma vez que recorre a sensores biométricos do Apple Watch. De forma abreviada, os objetivos principais são os seguintes:

- Estudar a integração do Apple Watch para recolher dados mais detalhados sobre os clientes durante os treinos, como é o caso dos batimentos cardíacos, da distância percorrida, do tempo, das repetições e das calorias perdidas;
- Automatizar o desempenho do PT:
 - Criação dinâmica de planos de treino baseado no objetivo físico do indivíduo, intensidade e batimento cardíaco;
 - Receber notificações se um cliente faltar e enviar um treino que poderá ser feito em casa;
- Introduzir um sistema de notificações através do Apple Watch para o cliente, dando a possibilidade de ler mensagens, lembretes de treino e informações durante o treino, para que o cliente também possa verificar todos os dados acima referidos, avisando-o quando alcança uma nova meta do exercício realizado.

A aplicação conta com um sistema de envio de mensagens entre o cliente e o PT e com a possibilidade de consulta da informação do cliente, como, por exemplo, a altura, o peso, os precedentes de doenças e o IMC. Existe ainda um calendário que pode ser consultado tanto pelo PT como pelo cliente e que permite verificar quando e quais os clientes que têm marcação

de sessão efetuada. Estas funcionalidades permitem consolidar todo o processo de trabalho de um PT numa só aplicação, tendo a vantagem de ser completamente portátil.

1.4 Resultados Esperados

Como resultados, espera-se alcançar os objetivos definidos na secção anterior para apoiar o PT, com o objetivo de facilitar e otimizar o seu trabalho. Mais concretamente, pretende-se desenvolver um protótipo da aplicação escalável, uma vez que é esperado que surjam novas funcionalidades e a possibilidade de desenvolver uma versão da aplicação para Android. Serão utilizadas as melhores práticas em desenvolvimento de software adquiridos no contexto da Empresa ITSector e do ISEP, com uso de **padrões de software adequados que facilitem a manutenção do projeto**, implementação de novas funcionalidades e a leitura e compreensão do código.

1.5 Análise de Valor

O valor apresentado será direcionado às partes interessadas que, neste caso, serão os PT's que se pretendam destacar num mercado em crescimento e em constante concorrência, com a esperança de agilizar o processo de geração de treinos específicos e acompanhamento dos seus clientes. Para isso, será necessário atingir os objetivos estabelecidos e esperar o feedback das partes interessadas.

Na Secção 3 será detalhada com maior granularidade a Análise de Valor, com recurso aos modelos *New Concept Development* (NCD), *Analytic Hierarchy Process* (AHP) e o Modelo Canvas, definindo-se os conceitos de Valor, Valor Percecionado e Valor para o Cliente e, por último, analisando a Rede de Valor da ITSector.

1.6 Estrutura

A presente dissertação está dividida nas seguintes secções: Introdução (Capítulo 1), Contexto (Capítulo 2), Análise de Valor (Capítulo 3), Estado de Arte (Capítulo 4), Análise (Capítulo 5) e Implementação (Capítulo 6). De seguida, será feita uma pequena explicação dos conteúdos de cada uma das partes:

- **Capítulo 1** – é a secção introdutória da dissertação, onde será feito um resumo do contexto do projeto, do problema, dos objetivos pretendidos, um excerto da análise de valor e da estrutura do projeto;
- **Capítulo 2** – este capítulo irá abordar com um maior detalhe o contexto e o problema;

- **Capítulo 3** – neste capítulo irá ser detalhada o aspeto mais teórico do negócio, isto é, o que é proposto para criar valor;
- **Capítulo 4** – será feita uma listagem detalhada da possível concorrência do mercado, definindo as oportunidades existente;
- **Capítulo 5** – será feita uma análise extensiva sobre o design da solução, ou seja, arquitetura, fluxogramas, design da aplicação e algumas referências a diagramas. Irá também ser feita uma observação aos conceitos mais teóricos do tema em questão, ou seja, definições de intensidade, exercícios, objetivos e zonas cardíacas. Por fim, será realizada uma análise às diversas abordagens ao projeto que estiveram em consideração, isto é, serão referidas as alternativas ou metodologias que poderiam ter sido adotadas;
- **Capítulo 6** – será explicada com maior granularidade a implementação das tecnologias utilizadas, esclarecendo de forma teórica a razão da sua implementação;
- **Capítulo 7** – por último, irá ser discutido todo o âmbito do projeto e serão referidas as conclusões desta dissertação e do projeto em questão;
- **Capítulo 8** – serão listadas as referências utilizadas para o desenvolvimento desta dissertação.

1.7 Publicação

O trabalho aqui apresentado foi publicado em (Ferreira, Coelho, & Nogueira, 2018) num trabalho que contou com a colaboração do professor Nuno Nogueira, docente de fisioterapia e especialista na área desportiva do IPSN/CESPU e ESS/IPP, assim como do professor Jorge Coelho, docente de informática no ISEP.

1.8 Sumário

O capítulo 1 (Introdução) tem como objetivo contextualizar o leitor, referindo o problema a ser resolvido e dando ênfase aos objetivos a ser alcançados para tal. É feita uma pequena introdução à análise de valor, referindo o valor que é esperado ser alcançado, sendo feita uma análise mais alargada no Capítulo 2. Finalmente, é referida a estrutura deste documento, caracterizando cada capítulo e o que cada um abordará.

Inserir página em branco apenas se necessário de modo a que o próximo capítulo comece numa página à direita

2 Contexto

Neste capítulo é apresentado o contexto do projeto e o problema que se esta a solucionar, de forma a explicar ao leitor onde o projeto se insere no mercado e como é esperado gerar valor às partes interessadas.

2.1 Contexto e Problema

2.1.1 Contexto

Devido a uma maior consciencialização das populações para os benefícios da prática regular de exercício físico e a maior preocupação com a imagem, a sua popularidade tem vindo a aumentar, e, por isso, segundo a AGAP, o mercado português **de fitness cresceu 13%**, isto referente ao ano de 2015, mas com tendência a este valor crescer (AGAP, 2016).

Há uma maior educação em relação à prática do exercício físico e dos benefícios que esta traz a curto e longo prazo. Segundo Sjøgaard, a prática de exercício físico com regularidade traz uma maior produtividade, reduz desordens musculares e problemas cardíacos (Sjøgaard, et al., 2016).

Para suplementar a prática de exercício físico, surgiu uma maior procura de alimentos ricos em proteína, fruta e mais verduras que, em conjunto, criam bons hábitos para se estar em forma e, essencialmente, para a sensação de bem-estar com a própria imagem. Segundo o Observatório da Nutrição e Alimentação em Portugal, **cerca de 65% da amostra do estudo realizado faz uma alimentação saudável e cerca de 90%** da amostra considerou que a alimentação é muito importante para a saúde (ONAP, 2018).

Como resultado, há um ambiente propício a um maior investimento por parte dos ginásios em maquinaria, espaço e, sobretudo, em professores ou nos chamados PT's, que pretendem obter uma maior quota do mercado, uma vez que este também cresceu para eles.

Deste modo é promovido um maior investimento por parte das grandes marcas já estabelecidas (Nike, Adidas, New Balance) (Nike, 2017) (Adidas, 2016) (Balance, 2011) ou até novas marcas a emergirem, como a Prozis (Nunes, 2015), com produtos mais direcionados para um estilo de vida saudável. Os ginásios também diversificam as estratégias para atrair cada vez **mais clientes pois existe uma maior competitividade entre os mesmos**, o que leva, por exemplo, ao aparecimento de descontos nas inscrições, à criação de pacotes de aulas especializadas e também a um maior número de aulas de grupo que promovem a atividade física e o relacionamento entre PT e cliente.

Outro dos mercados onde houve um grande investimento foi o das aplicações móveis para *Smartphones*, com aplicações de auxílio ao treino, como Nike + Training Club (Apple, Nike Training Club, 2017) e Runkeeper (Apple, Runkeeper, 2017), que, de certa forma, substituíram o PT. Este auxílio pode ir desde o plano de treino até à gestão da alimentação. Segundo o novo relatório da Flurry Analytics (Klotzbach & Kesiraju, 2017), as aplicações para saúde e *fitness* estão a crescer com um ritmo mais acelerado até agora do que o mercado global de aplicações móveis em 2014. A empresa de análise verificou os dados de mais de 6800 aplicações móveis para iPhone e iPad e constatou **um crescimento de 62% nas aplicações** deste tipo nos últimos seis meses, enquanto que as restantes aplicações tiveram uma média de crescimento de 33%. Comparando, as aplicações para o bem-estar físico cresceram 87% mais rápido do que qualquer outra aplicação.

Tudo isto leva a crer que existe uma grande vontade por parte dos vários mercados acima mencionados em **apostar no *fitness* e no exercício físico**, pois existe um grande interesse das pessoas por esta área. No que diz respeito às partes interessadas desta solução, ou seja, os PT's, isto proporciona-lhes uma oportunidade de maior visibilidade e de crescimento a nível profissional.

2.1.2 Problema

Como foi atestado, existe um grande crescimento do mercado do *fitness* e uma oportunidade para o PT se tentar destacar no mesmo. No entanto, este também é um dos fatores que, de certa forma, veio colocar em risco a posição do PT, ou seja, o facto de existir um **número crescente de aplicações *fitness* gratuitas para qualquer pessoa** que explicam e exemplificam ao utilizador como executar os exercícios e dispõem de um vasto número de exercícios disponíveis onde, posteriormente, é possível gerar um plano de treino consoante a zona de corpo ou o tipo de treino pretendidos.

Segundo Cameron Lister (Lister, West, Cannon, Sax, & Brodegard, 2014), a criação do conceito de *Gamification* nas aplicações promove a socialização entre os utilizadores de uma forma interativa e divertida, oferecendo recompensas pelos objetivos atingidos. Este conceito de motivação está presente em muitas aplicações móveis que aliciam os utilizadores e que, de alguma forma, substituem ou tentam substituir o PT.

Com a quantidade de informação disponível na internet, o papel de um PT torna-se cada vez mais irrelevante no mercado ou, no mínimo, ameaçado. Há a **necessidade de tomar partido das tecnologias** e, por exemplo, centralizar toda a informação referente aos clientes num dispositivo móvel, automatizando o seu trabalho para reforçar a sua posição no mercado, de forma a justificar o seu recurso enquanto profissional de *fitness* e exercício físico.

Existe ainda o **fator económico**, uma vez que este acaba por impossibilitar muitas pessoas de aceder aos benefícios de um PT e de perceber as vantagens que o seu trabalho pode trazer para o seu treino. Desta forma, é necessário dar razões às pessoas para utilizarem os serviços de um PT, dando-lhes mais valor através de soluções tecnológicas.

Perante a pesquisa feita no mercado *mobile*, existem muito **poucas aplicações que auxiliam** o PT neste sentido, sendo de extrema importância delegar uma nova forma de gerir o seu negócio e de manter a relação com os seus clientes. Um dos nossos objetivos é precisamente que o PT possa centralizar toda a informação dos seus clientes e automatizar/simplificar todas as fases do seu trabalho.

Desta forma, o PT estará apto para, tirando partido das novas tecnologias, **ver a sua posição reforçada e não desvalorizada**. É expectável que a aplicação seja utilizada pelos PT's de forma independente, o que a tornará uma ferramenta pessoal, de forma a substituir o uso de papel e múltiplas aplicações.

Tudo isto proporcionará novos desafios, na medida em que será necessário desenvolver uma aplicação móvel para iOS com integração do Apple Watch e dos seus sensores, tirando o melhor partido de toda a informação a que temos acesso.

2.1.3 Restrições Existentes

Existiram várias restrições no início deste projeto, estando algumas delas descritas de seguida.

A base da aplicação já estava desenvolvida para a tecnologia iOS pela ITSector, que se tratava de um teste ao mercado *fitness*. Devido à fraca adesão, **não houve interesse em desenvolver para outras plataformas como Android ou até mesmo Windows Phone**. Isto acabou por ser uma vantagem, uma vez que é uma tecnologia bastante conhecida na ITSector e, caso surgisse algum problema, haveria sempre alguém a quem recorrer.

Em termos monetários não existiu qualquer restrição, uma vez que todos os dispositivos necessários (iPhone e Apple Watch) já existiam na ITSector. Da mesma forma, a base de dados e o servidor a ser utilizados já existiam, não tendo exigido qualquer investimento extra.

No que toca a restrições temporais, não houve a possibilidade de **trabalhar neste projeto a tempo inteiro**, o que originou atrasos em alguns desenvolvimentos na parte técnica.

3 Análise de Valor

A análise de valor neste ponto é referente ao Problema apresentado na secção anterior e visa criar valor na oportunidade existente. Este capítulo será detalhado, referindo alguns modelos (Canvas, AHP, NCD) no sentido de reforçar esta análise.

3.1 Introdução

Esta análise tem como objetivo identificar oportunidades de aumentar o valor das funções de produtos ou serviços de acordo com as necessidades dos clientes alvo, assim como reconhecer as funções de valor dos produtos e processos, quantificando os seus benefícios e custos relativos, retirando o foco de seus componentes (Usirono, 2015).

Para se desenvolver novos e inovadores produtos, é necessário que a instituição esteja disposta a correr riscos, uma vez que o produto inovador pode fracassar e danificar o nome da instituição, da mesma forma que podem existir danos severos no poder económico da empresa. Assim, existe a necessidade de correr o mínimo risco possível. Desta forma, segundo o processo de inovação de Reinertsen, existem três subprocessos do processo de inovação (Teza, Miguez, Fernandes, & Dandolini, 2013):

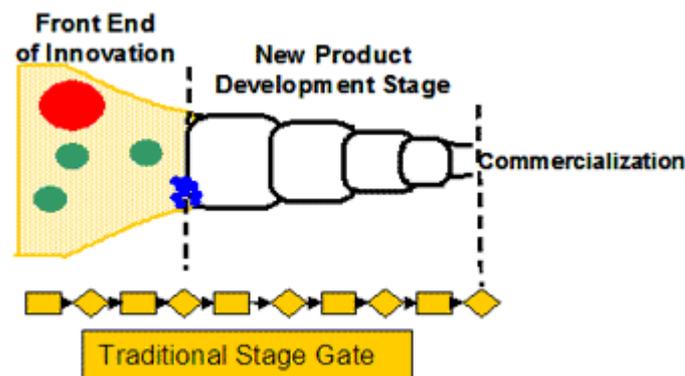


Figura 1 - Processo de Inovação

- **Fuzzy Front End (FFE)** – também descrito como “*Stage 0*”, é quando se estudam as oportunidades e é criado um conceito antes de ser algo desenvolvido;
- **Desenvolvimento de novos produtos** – fase onde já estão estabelecidos todos os objetivos e onde há foco no desenvolvimento do produto/serviço;
- **Comercialização** – fase onde se divulga o produto ao público.

Nesta dissertação será apenas abordado o novo conceito de desenvolvimento de Peter Koen (NCD), uma vez que foi desenvolvido a partir da necessidade de identificar com maior clareza o *Fuzzy Front End*, o denominado *Front end of innovation* (FEI).

3.2 The new concept development model (NCD)

Segundo Peter Koen, este modelo (NCD) define uma linguagem comum e uma visão holística do FEI. Este modelo é ainda subdividido em três componentes que influenciam todo o processo de inovação (Koen, What is the New Concept Development (NCD) model?, 2018):

- **Engenho/centro do modelo (Engine):** promove “força” para o *front end* da inovação, é a parte fulcral do negócio, onde tem em conta a visão, estratégia e cultura que o move;
- **Parte interior do modelo (Inner part):** define os cinco elementos da parte interior, sendo eles: identificação de oportunidades, análise de oportunidades, geração e aperfeiçoamento de ideias, seleção de ideias e desenvolvimento do conceito e da tecnologia;
- **Parte exterior do modelo (External environmental factors):** é constituída por todos os fatores externos ao modelo, ou seja, incontrolláveis.

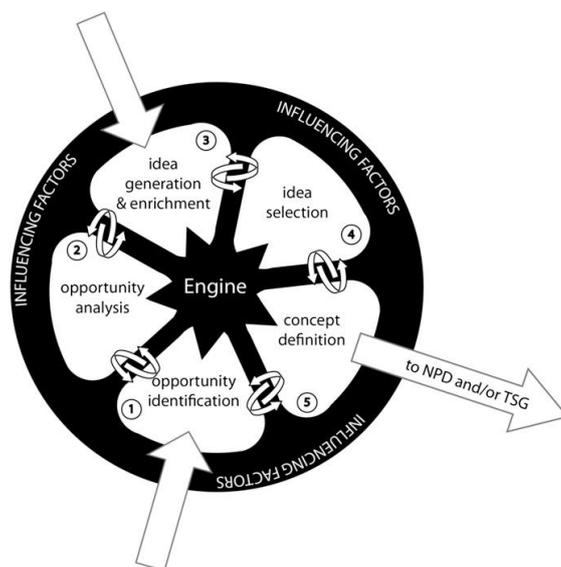


Figura 2 - New Concept Development Model

O processo representado no modelo da Figura 2 (Koen, What is the New Concept Development (NCD) model?, 2018) inicia-se na identificação de uma oportunidade ou na geração e aperfeiçoamento de uma ideia (Koen, Bertels, & Kleinschmidt, Managing the Front End of Innovation—Part II, 2014). Essa oportunidade ou ideia pode interagir com todos os elementos do modelo e com os fatores de influência (interiores e exteriores) impulsionados pelo motor/centro do modelo. O modelo circular permite ter a percepção que as ideias e oportunidades devem fluir entre todos os cinco elementos acima referidos.

Assim, o modelo tem dois pontos de início, mas só um de saída, sendo esse ponto a **ligação entre o processo de desenvolvimento formal** e a atividade de desenvolvimento de conceito e tecnologia (Teza, Miguez, Fernandes, & Dandolini, 2013).

Assim, na aplicação em questão a entrada é a identificação de uma oportunidade, uma vez que esta já está desenvolvida pela instituição. Desta forma, existe a oportunidade de auxiliar o PT na prática do seu trabalho, criando algo diferenciado do que já existe no mercado.

3.2.1 Parte interior do modelo (Inner Part)

Como foi referido, nesta secção será feita a caracterização dos cinco elementos chave da parte interior do modelo NCD no âmbito do documento e da proposta de valor.

3.2.1.1 Identificação da Oportunidade

Conforme o que foi referido na Secção 2.1.1, existe, por parte das populações, uma maior consciencialização no que toca ao bem-estar e aos benefícios do exercício físico para a saúde.

Consequentemente, isto originou um grande investimento por parte das marcas de desporto e *fitness* que pretendem tirar partido deste crescimento (Bain, 2017), dando possibilidade de criação de novas marcas e produtos ligados ao mercado, como a **Prozis, linhas de roupa, calçado e outros acessórios**.

Este investimento também abrangeu os **ginásios**, uma vez que a sua popularidade tem vindo a crescer, aumentando o número de ginásios e levando a uma maior competitividade neste setor. Segundo a AGAP, o mercado de **fitness português cresceu 13%**, com tendência crescente deste valor (Publicidade, 20016).

Como tal, existe uma maior procura de ginásios por parte das pessoas que pretendem um **estilo de vida saudável**. Isto permite ao PT uma maior oportunidade de negócio, no entanto, existirá também uma maior concorrência entre vários PT's. Assim, existe uma maior exigência por parte dos clientes no que toca à utilização dos serviços do PT, o que leva os PT's a estarem ocorrentes das novas metodologias de trabalho para que possam aliar os seus conhecimentos técnicos aos avanços tecnológicos de forma a se destacarem.

Como não existem muitas aplicações de auxílio ao PT, surgiu a oportunidade da ITSector apostar no **desenvolvimento de uma solução direcionada ao mesmo**. O seu objetivo seria permitir gerir todos os seus clientes e agrupar os dados das últimas sessões para fortalecer a posição do PT. Desta forma, o cliente sentirá que o PT está atento a todos os detalhes do seu corpo e do seu desempenho.

3.2.1.2 Análise da oportunidade

Como suporte da oportunidade definida, foram realizadas pesquisas para identificar o crescimento do mercado *fitness* e qual dos *wearables* seria mais propício desenvolver, uma vez que existe uma vasta gama no mercado.

Houve a **hipótese de desenvolver a aplicação móvel também para Android**, uma vez que estaria acessível a um maior número de PT's, mas, tendo em conta que a base da aplicação estava desenvolvida em iOS e que existia algum conhecimento desta tecnologia, a opção mais lógica foi **manter o desenvolvimento em iOS**.

Discutiram-se os custos e riscos por parte da instituição, assim como os contratempos que poderiam surgir. Tendo em conta que existia bastante conhecimento de iOS e Apple Watch, não houve qualquer preocupação quanto aos possíveis riscos. Quanto a custos, seriam reduzidos, uma vez que todo o material necessário (iPhone e Apple Watch), a base de dados e o *Middleware* estavam disponíveis.

3.2.1.3 Geração e aperfeiçoamento de ideias

Após a análise da oportunidade, discutiu-se com indivíduos que trabalham na área, com pessoas com conhecimento abrangente do mercado e com a instituição. Surgiram assim várias ideias e evoluções de outras aplicações que tornariam a solução atual muito mais completa e versátil, dando algumas funcionalidades aos clientes do PT (rede social dos clientes do PT, *gamification* dos treinos). Teriam ainda a capacidade de estudar o batimento cardíaco, usando o HRV (variabilidade do batimento cardíaco) (Campos, 2017) e o desempenho, através do VO2max (Quinn, 2018).

No entanto, o objetivo inicial era **focalizar a aplicação no acompanhamento dos clientes** que não têm objetivos de alta competição (VO2max), ou seja, nos clientes que utiliza os serviços do PT para manter uma forma física saudável e assim **fortalecer a posição do PT ao agilizar os processos de treino com recurso a *wearables***.

Esteve também em discussão qual *wearable* usar, uma vez que existem várias ofertas. Mostrou-se de extrema importância utilizar dispositivos com sensores precisos (como, por exemplo, o monitor de batimentos cardíacos) para fornecer a informação mais credível possível aos clientes do PT.

Existiu ainda a possibilidade de utilizar um *chip* para colocar na camisola. Através de uma API, seria possível aceder à informação dada por vários parâmetros, nomeadamente a temperatura do corpo e o batimento cardíaco. Esta ideia foi descartada, uma vez que criaria dependência de uma instituição terceira e necessitaria de alguma burocracia no processo de comercialização da aplicação.

3.2.1.4 Seleção de ideias

Com base na reflexão feita, focou-se no monitor de batimentos cardíacos do *smartwatch* Apple Watch, uma vez que seria uma extensão natural à aplicação móvel que está a ser desenvolvida para iOS. Ainda segundo o estudo (Shcherbina, et al., 2017), este monitor fornece os batimentos cardíacos de forma mais precisa, comparativamente com os restantes *smartwatches* e *fitness bands* disponíveis no mercado.

Conclui-se que o PT necessita de uma plataforma que o permita gerir todos os seus clientes, usando as mais recentes tecnologias de forma ágil no seu dia-a-dia, levando ao contentamento dos seus clientes.

3.2.1.5 Desenvolvimento do conceito

O conceito já foi aprovado pela instituição que avaliou a fiabilidade do projeto, tendo conhecimento da mão de obra a ser utilizada para o desenvolvimento, do financiamento e da estratégia de marketing. É importante referir que a aplicação móvel **já se encontra no mercado**, estando numa fase madura do seu ciclo de vida e sendo necessário acrescentar valor à mesma.

Existe a possibilidade de ser criada uma campanha de marketing com o objetivo de captar novamente a atenção do PT e criar expectativas aos clientes.

3.2.2 Parte exterior do modelo (*External environmental factors*)

Para se efetuar corretamente esta análise, serão utilizados os fatores externos da análise **PEST** (Política, Económica, Social e Tecnológica) abordados na unidade curricular de EINOV. Ter-se-ão também em conta o Contexto e as Oportunidades que já foram referidas e que estão diretamente relacionadas com os seguintes fatores:

- **Fatores Políticos** – terá pouca relevância para o contexto da aplicação móvel em discussão, uma vez não cria nem riscos nem oportunidades para o projeto;
- **Fatores Económicos** – existe um aumento do mercado *fitness* e de ginásios em Portugal, o que gera uma maior oportunidade para se apostar neste setor;
- **Fatores Sociais** – o fator principal é, sem dúvida, o facto de existir um maior cuidado pelo bem-estar físico e pelo exercício, mas também, segundo a Statista há um aumento na utilização de *smartphones* e no download de aplicações móveis que chegou aos 180 biliões (só na App Store) (Statista, Cumulative number of apps downloaded from the Apple App Store from July 2008 to June 2017, 2017), o que leva a uma maior facilidade de adesão neste setor;
- **Fatores Tecnológicos** – há um aumento no investimento por parte das grandes marcas de desporto, *fitness* e até de *smartphones* em *wearables* que têm como visão acompanhar o indivíduo na atividade física, assim como criar uma oportunidade para auxiliar os PT's. Isto

pode também ser uma ameaça para os mesmos, uma vez que estes *wearables* podem diminuir o impacto do PT no mercado do *fitness* e do exercício físico.

3.3 Valor, Valor Percecionado e Valor para o Cliente

Neste subcapítulo será feita uma análise aos conceitos de Valor e a quem se destina, ao Valor percecionado e, por último, ao Valor para o cliente.

3.3.1 Valor

Valor é a expectativa do consumidor referente, por exemplo, à qualidade e robustez de um produto, ou, caso seja um serviço, à fiabilidade e rapidez do mesmo em relação à sua quantia monetária.

Durante uma compra, o consumidor **analisa e avalia o valor do produto ou serviço que a empresa oferece e o valor dessa transação**. É de referir que este valor pode não ser só monetário, mas também temporal ou de esforço. Desta forma, para haver negócio terá de ser criado valor, uma vez que apenas com valor é que o consumidor compra repetidamente o produto/serviço (Ikeda & Oliveira, 2005).

No caso da aplicação móvel em questão, terá de haver valor para o PT. Tendo em conta que se trata de um produto não tangível, é necessário que a aplicação seja fiável, rápida e que realize as tarefas de acordo com o valor proposto.

3.3.2 Valor Percecionado

Referido também como valor percebido (*perceived value*), refere o valor que o consumidor percebe e perceciona. Pode não ser o valor que um outro consumidor percebe, da mesma forma que pode não ser o valor que a empresa responsável pelo produto/serviço tenha percebido (Jorge & Simões, 2010).

Portanto, pode-se dizer que o valor percecionado **varia de consumidor para consumidor e depende da expectativa do mesmo**.

No caso da aplicação que este documento detalha, poderá haver uma boa percepção por parte do consumidor (PT), uma vez que agilizará o seu trabalho e terá a vantagem da portabilidade. Apesar disso, poderá haver alguma curva de aprendizagem ou a necessidade de habituação a esta nova metodologia de gerenciamento de trabalho.

3.3.3 Valor para o Cliente

Existem várias definições para o Valor para o Cliente. Isto acontece porque este é um espectro **altamente dinâmico** que se deve às características que o consumidor atribui a um produto/serviço, características essas que podem estar constantemente em mudança.

Por isso, a definição usada nesta dissertação será a de Churchill e Peter (2000): “Valor para o cliente é a diferença entre as percepções do cliente quanto aos benefícios e quanto aos custos da compra e uso de produtos e serviços.” (Martins, 2001)

Neste caso, o PT terá de medir se o custo da utilização da aplicação está de acordo com as suas expectativas e necessidades e se, realmente, existem vantagens em usar a aplicação, comparativamente com os métodos atuais que estes utilizam.

3.4 Benefícios e Sacrifícios para as Partes Interessadas

Como nos conceitos referidos na Secção 3.3, serão agora mencionados os benefícios e os sacrifícios relevantes para o mercado alvo/cliente que, neste caso, é o PT.

Tabela 1 - Tabela de Benefícios/Sacrifícios

	<i>Produto/Serviço</i>	<i>Relação</i>
<i>Benefícios</i>	Aumento na produtividade do PT; Melhor acompanhamento dos seus clientes.	Aumento do contentamento dos clientes; Maior destaque no mercado onde existe muita concorrência.
<i>Sacrifícios</i>	Curva de aprendizagem; Criar hábitos de trabalho.	Possível descontentamento dos clientes; Provoca o oposto do objetivo pretendido.

3.5 Proposta de Valor

A solução apresentada pretende solucionar o problema relacionado com a **inexistência de uma aplicação fiável para gerir e automatizar o trabalho do PT**. O problema referido está explicado na secção 2.1.2 com maior detalhe.

O objetivo é desenvolver uma aplicação fiável e de fácil acesso, com recurso às capacidades do Apple Watch, de forma a gerar valor às partes interessadas (PT). A solução será detalhada com mais pormenor técnico ao longo deste documento.

3.6 Modelo Canvas

Em qualquer negócio é necessário estudar a forma como a organização se deve estruturar para gerar e capturar valor. A estes “estudos” dá-se o nome de **Modelo de Negócio**, tendo em conta que definem a estrutura económica, estratégica ou de operação da empresa, assim como a forma como a empresa vai resolver um problema às partes interessadas, usando um meio que gere valor (Pereira, 2016).

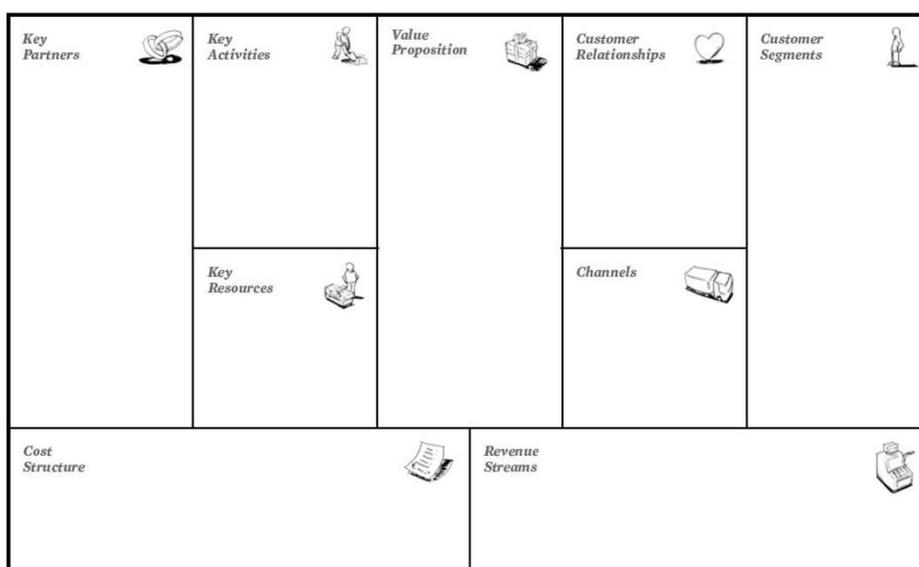


Figura 3 - Modelo Canvas

Uma das formas mais usuais de se realizar o plano é através do chamado **Modelo Canvas**, uma vez que é sintético e visual, logo, bastante participativo. Este modelo está dividido em dois *stages*: o **Back Stage** e o **Front Stage**, que, por sua vez, estão divididos em vários subpontos, os quais serão explicados a seguir (EINOV, 2017):

- **Back Stage:**
 - **Parcerias chave:** quem são os parceiros chaves do negócio? Quem são os fornecedores principais?
 - **Atividades chave:** que tipo de atividades chaves a proposta de valor necessita? Canais de distribuição? Relações com os clientes?
 - **Recursos chave:** que tipo de recursos chave a proposta de valor necessita? Canais de distribuição? Relações com os clientes?
 - **Perspetiva de custos:** quais são os custos mais importantes para o negócio? Quais os recursos chave mais caros?
- **Front Stage:**

- **Proposta de valor:** que valor irá ser gerado para os clientes? Que necessidades se deve satisfazer?
- **Canais:** que canais devem ser utilizados para se chegar a cada segmento de mercado? Quais deles trabalham melhor?
- **Mercado alvo:** para quem irá ser criado valor? Quem são os melhores clientes para o negócio?
- **Relações com os clientes:** que tipo de relação que cada segmento de mercado está à espera da empresa? Quais relacionamentos devem ser estabelecidos?
- **Perspetivas de receita:** por que valor os clientes estão dispostos a pagar? O que eles atualmente pagam?

De seguida serão detalhados todos os pontos relativos à aplicação a ser desenvolvida no âmbito do documento.

3.6.1 Mercado Alvo (Customer Segments)

A criação da aplicação terá como objetivo servir o PT, auxiliá-lo no acompanhamento dos seus clientes e automatizar o processo de criação de treinos apropriados para cada cliente.

Indiretamente, os ginásios também poderão ter um aumento de vendas, uma vez que, muitas vezes, os PT's pertencem a algum ginásio. Se estes tiverem a aplicação, haverá um possível aumento da sua produtividade e conseqüentemente, um maior contentamento por parte dos clientes e um aumento das receitas obtidas pelos ginásios.

3.6.2 Proposta de Valor (Value Propositions)

Com este produto é proposto aumentar a rentabilidade dos PT's. Estes poderão centralizar toda a informação sobre os clientes e automatizar os processos de geração de treinos específicos para cada um deles, recorrendo ao Apple Watch que fornece informação como o batimento cardíaco, as calorias perdidas, a distância percorrida.

A portabilidade da aplicação torna muito mais simples a gestão do trabalho, não existindo a necessidade de documentos extra.

3.6.3 Canais (Channels)

Os canais principais para a comunicação serão as redes sociais, onde irá ser feita publicidade à aplicação, assim como nos ginásios onde atuam a maior parte dos PT's. Os canais de distribuição passarão pelo download na App Store da Apple, onde qualquer PT poderá descarregar a aplicação, efetuando o pagamento da mesma.

3.6.4 Relações com os clientes (Customer Relationships)

A relação com os clientes será feita maioritariamente através da aplicação, da chamada por voz ou via email, de forma imediata e dedicada, de modo a cativar o cliente a utilizar a aplicação.

É ainda de esperar que a utilização da aplicação seja fácil e amigável para o PT, não fazendo com que se sinta frustrado ou com sentimento de falta de controlo sobre as suas ações.

3.6.5 Perspetivas de Receita (Revenue Streams)

A principal fonte de receita do produto será através do download pago da aplicação base por 30 euros, com a possibilidade de criação de um pacote Premium que requer uma taxa extra, estando a ser planeada uma subscrição paga de 10 euros por mês. Este pacote traria um maior número de exercícios que poderiam abranger novos desportos e novas funcionalidades, como é o caso da criação de estudos de performance baseados nos resultados do cliente em questão. Relativamente ao pacote Premium, como seria uma subscrição, seria expectável o trabalho contínuo na aplicação após o seu lançamento.

3.6.6 Recursos Chave (Key Resources)

De modo a cumprir com o que é proposto, o produto tem de satisfazer as necessidades do PT de forma simples e intuitiva, com suporte a dúvidas e a aparecimento de eventuais erros.

Existe obrigação de oferecer a melhor experiência possível, uma vez que se trata de uma aplicação paga e o valor tem de ser justificado. Como existem muitas variáveis para o bom funcionamento da aplicação com o Apple Watch como, por exemplo, a distância entre o telemóvel e o relógio, a rede disponível no local e o bom posicionamento do relógio no pulso do cliente, é fulcral testar e otimizar a aplicação para garantir o melhor resultado.

3.6.7 Atividades Chave (Key Activities)

A principal atividade é fornecer uma aplicação ao PT que automatize e facilite o gerenciamento dos seus clientes. Usando sistemas de mensagens entre o PT e os seus clientes e recorrendo ao Apple Watch e aos seus sensores será extraído o maior número de informação do cliente durante a sessão de treino (batimento cardíaco, calorias perdidas, distâncias, etc.).

Com estes dados, a aplicação pode gerar treinos automáticos consoante a zona do corpo ou intensidade, excluindo assim o processo de elaboração de um plano de treino específico para cada cliente. Para além disso, será possível verificar o histórico do utilizador em termos de peso, batimento cardíaco e até calorias perdidas.

3.6.8 Perspetiva de Custos (Cost Structure)

Relativamente a este tópico, é perspetivado existirem custos no que toca ao desenvolvimento da aplicação, assim como para manter a aplicação na App Store e no processo de pós lançamento com desenvolvimento de novas funcionalidades. Existirão também custos relacionados com o servidor onde ficará alojada toda a informação dos PT's.

Para além da aplicação será necessário investir numa equipa de marketing e publicidade, uma vez que é necessário alcançar o maior número de PT's e publicitar a aplicação.

3.6.9 Parcerias chave (Key Partners)

Como parcerias chave, seria necessário entrar em contacto com várias empresas de *fitness* e de alimentação (como Prozis, Manz e GymPas) e ginásios, uma vez que são marcas que estão bastante presentes na cultura e no dia-a-dia do PT e que, de facto, representam valor para o mesmo.

3.7 Rede de Valor

Segundo Verna Allee, a Rede de Valor ou *Value Network* é um conjunto de papéis, interações e relações que geram valor económico e social. Assim, qualquer organização, instituição ou atividade poderá ser considerada uma Rede de Valor.

A análise de uma Rede de Valor define papéis na organização e como estes interagem na rede, tornam produtos/serviços intangíveis visíveis e negociáveis, podendo modelar fluxos complexos de valor e de recursos humanos (Allee, 2012).

“People naturally network as they work so why not model itself as network” (V.Allee)

Com base na afirmação e no que foi referido nesta secção, é necessário que modelar o trabalho de todos os departamentos em Rede, de forma a organizar os recursos tangíveis e não-tangíveis de uma organização ou atividade.

Neste caso, a aplicação móvel está integrada no âmbito da empresa ITSector e estabelece a integração entre vários departamentos (Design, Desenvolvimento e Marketing), o que gera uma troca de ideias e informação entre os mesmos, tornando a aplicação mais coesa e completa.

No entanto, existe uma rede que ajudaria na valorização da aplicação e na manutenção da mesma, rede essa que seria a dos parceiros que, neste caso, seriam os ginásios e os PT's que utilizariam a aplicação. Esta seria a mais importante, uma vez que facilitaria a comunicação entre a ITSector e o nosso mercado alvo.

3.8 Modelo AHP

O Modelo Analytic Hierarchy Process (AHP) é um modelo ou ferramenta que permite lidar com a escolha de decisões complexas e que ajuda o decisor a definir prioridades e escolher a melhor decisão. Este modelo reduz as decisões complexas numa série de comparação de critérios e, de seguida, sintetiza os resultados de forma a capturar os aspetos subjetivos e objetivos de uma decisão (Mocenni).

No contexto da aplicação móvel será utilizado este modelo para intervir aquando da tomada de decisão. Para tal, serão definidos os seguintes critérios:

- **Qualidade e credibilidade de resultados** – os resultados obtidos e apurados pela aplicação terão um grau de fiabilidade alto, uma vez que se trata de informação sobre um cliente. Caso esta case seja incorreta, pode aumentar o descontentamento do mesmo;
- **Facilidade de uso** – a aplicação terá de ser de fácil utilização, responsiva e simplista;
- **Adaptativa** – a aplicação terá de se adaptar às necessidades do PT, ou seja, deverá estar sempre ocorrente do feedback do mesmo para eventuais novas funcionalidades.

Estes critérios serão avaliados aquando do desenvolvimento da aplicação móvel com o objetivo de focar a tomada de decisão e gerar maior valor para o PT.

4 Estado de Arte

Com o recente crescimento do mercado do *fitness*, surgiu um maior número de aplicações móveis relacionadas com esta área que permitiram auxiliar o utilizador na prática de exercício físico, criando planos de treino e demonstrando a execução do exercício.

Este acompanhamento aumentou a vontade do utilizador relativamente ao registo de todos os dados associados a um treino, de modo a medir a sua capacidade física. Estes dados podem ir desde a distância percorrida até às calorias perdidas. Este registo levou à criação de estudos de performance onde só se poderiam inserir os dados manualmente, sendo estes obtidos através de máquinas profissionais que, uma vez que são bastante dispendiosas, não estão ao alcance de qualquer pessoa.

Surgiu então a oportunidade de se criar os chamados “*companion*” *gadgets* mais acessíveis e de uso pessoal que permitem recolher informação do utilizador. Estes *gadgets* são simples de usar e têm ligação com os *smartphones* atuais, o que agiliza a gestão de informação.

Inicialmente, estes dispositivos começaram por ser substituídos pelos *smartwatches* que, para além de terem funcionalidades de rastreio de passos, repetições de um exercício e batimento cardíaco, têm uma panóplia de funcionalidades associadas a *smartphones*, como atender chamadas, escrever mensagens por voz, etc. Por sua vez, surgiram as chamadas “*fitness bands*”. Apesar de bastante semelhantes aos *smartwatches* em termos de aparência, estas são mais acessíveis e têm como único propósito a monitorização da performance desportiva do indivíduo, podendo, posteriormente, enviar a informação para o *smartphone* associado.

Nas secções seguintes será feita uma análise mais detalhada dos dispositivos disponíveis no mercado para auxílio à prática de exercício físico e as suas aplicações associadas.

4.1 *Smartwatches*

Com a entrada dos *smartwatches* no mercado dos relógios e *wearables* já estabelecido, seria normal de esperar que estes dificilmente conseguissem superar os relógios ditos normais em termos de vendas. No entanto, a própria Apple provou o contrário ao anunciar que o Apple Watch se tornou no relógio mais vendido no globo, ultrapassando marcas como Rolex, Fóssil e Omega (Lora, 2017). Isto prova que existe um grande interesse por parte dos consumidores relativamente à utilização dos *smartwatches*. Devido a isso, as restantes marcas, vendo o sucesso do Apple Watch, estão a tentar obter uma quota do mercado dos *smartwatches*.

De seguida serão apresentadas com maior detalhe as características de alguns dos *smartwatches* mais vendidos e disponíveis em Portugal, segundo a Canalys (Alto, Shangai, & Reading, 2017).

4.1.1 Apple Watch

Como foi referido anteriormente, este é, neste momento, o relógio número 1.

Contagiado em parte pelo sucesso de vendas do iPhone, este *smartwatch*, já existente na versão 5, passou a integrar ligação LTE. Com tecnologia avançada, como processador dual core, este novo Apple Watch é 70% mais rápido. As melhorias verificam-se também noutras áreas, nomeadamente no Wi-Fi que é 85% mais rápido e 50% mais eficiente e no cartão SIM que é eletrónico. A assistente inteligente pessoal de reconhecimento de voz (Siri) também melhorou, fruto deste novo processador, podendo finalmente interagir com o utilizador. Para integrar a ligação com a rede móvel, o próprio ecrã do Apple Watch funciona como antena (Apple, Apple Watch, s.d.).

Com o novo WatchOS, o sistema operativo no Apple Watch, as aplicações de obtenção de métricas do utilizador foram melhoradas, uma vez que agora obtêm informações como a variabilidade do ritmo cardíaco, as repetições dos exercícios e as listas de reprodução de músicas durante a sessão de treino. Possuem ainda uma aplicação de respiração chamada *Breath* (Apple, Breathe App, 2019), com o objetivo de fazer o utilizador relaxar durante 1 minuto através da respiração profunda.

Outra aplicação introduzida a partir do modelo Série 4 foi a oportunidade de efetuar um ECG para monitorizar o estado do coração do utilizador, dando alertas de quando este está em grande esforço cardíaco (Apple, Efetuar um ECG com a app ECG no Apple Watch Series 4 ou posterior, 2019).

É notório o esforço da Apple em construir um relógio que tem um enorme leque de funcionalidades direcionadas ao *fitness* e ao bem-estar (Apple, WatchOS, s.d.).

4.1.2 Samsung Gear S3

Com as oportunidades existentes no mercado dos *smartwatches*, a Samsung decidiu criar o seu próprio dispositivo.

Este *smartwatch*, surgiu com dois modelos disponíveis, a versão Classic e a versão Frontier. O Gear S3 Frontier é direcionado ao público mais ativo, uma vez que possui funcionalidades de treino, rastreio de batimento cardíaco e análise do sono. Para além disso, conta com GPS e navegação com comandos de SOS. Tal como o Apple Watch, tem também gestor de chamadas e envio de mensagens (Samsung, Samsung Gear S3 Frontier, s.d.).

Para além disso, o design do relógio em si é bastante semelhante a um relógio tradicional, tendo um ecrã redondo metalizado. É aqui que o Samsung Gear S3 se distancia visualmente do Apple Watch, uma vez que este tem um ecrã quadrado.

É de referir ainda que a loja de aplicações do Samsung Gear S3 é bastante mais extensa que a do Apple Watch, tendo em conta que possui algumas aplicações pagas e outras gratuitas (Coelho, Pplware, 2017), estando entre elas a meteorologia, o Spotify (Spotify, 2018), jogos e até mapas (Samsung, Samsung Gear S3 Apps list, s.d.).

É importante fazer uma referência ao cardiofrequencímetro, já que, segundo o artigo de Anna Shcherbina (Shcherbina, et al., 2017), o Samsung Gear é o *smartwatch* com maior margem de erro na obtenção do batimento cardíaco (cerca de 6.8% de margem de erro comparativamente com 2.0% do Apple Watch). Neste domínio, a precisão dos resultados é importante, principalmente quando estes condicionam o que é exigido ao utilizador e estando em causa a sua saúde.

4.1.3 TAG Heuer Connected Modular

Ao contrário dos exemplos anteriores, o *smartwatch* TAG Heuer Connected Modular é dos primeiros *smartwatches* a serem lançados por uma marca de relógios tradicional.

Este *smartwatch* tem um formato bastante similar a um relógio tradicional da marca e permite executar o sistema operativo Android Wear. Assim, possui uma vasta gama de funcionalidades que, por exemplo, o Samsung Gear S3 também detém, mas com o acréscimo de estar equipado com sensores de vento lateral e monitorização da previsão atmosférica.

É notório que o TAG pretende criar um relógio digital que se assemelhe a um relógio tradicional, mas com funcionalidades que auxiliem o dia-a-dia do utilizador.

Este dispositivo dispõe ainda de *Wi-fi*, Bluetooth, giroscópio, microfone, mas não tem antena, o que é uma desvantagem em relação à concorrência. Com a aplicação móvel exclusiva, facilmente se envia a informação do telemóvel em questão para o *smartwatch* e vice-versa.

Uma das inovações deste *smartwatch* é a sua personalização, uma vez que uma pessoa pode facilmente alterar a bracelete e a carcaça do relógio (Heuer, s.d.).

A maior desvantagem deste *smartwatch* é o seu preço, uma vez que requer um maior investimento.

4.2 Fitness Bands

Recentemente, surgiram as *Fitness Bands* ou Activity Trackers. Orientadas para o mercado do *fitness* e de exercício físico, estas são um subtipo dos *smartwatches*, uma vez que são dispositivos mais especializados na monitorização da atividade física do indivíduo.

Estes dispositivos tiveram a sua génese numa parceria entre a Apple e a Nike. Em conjunto, desenvolveram um dispositivo para colocar na sapatilha cujo objetivo seria calcular os passos

dados, obtendo assim a distância percorrida. Quando ligado ao iPod, era possível visualizar os dados obtidos pelo indivíduo. (Ridgers, 2016)

Este conceito evoluiu para o que conhecemos hoje como o Apple Watch. No entanto, o mercado para as *smartbands* teve continuidade, uma vez que os *smartwatches* requerem um maior investimento. O seu preço acessível e o seu crescente foco no exercício físico são os principais aspetos que proporcionam uma grande aderência a estes dispositivos.

Assim, será feita uma análise mais extensa às características das *smartbands* mais utilizadas no momento, segundo a AndroidPit e de acordo com alguns artigos publicados (AndroidPit, 2019) (Salutes, 2017) (Ridgers, 2016) (Shcherbina, et al., 2017) (Stahl, An, Dinkel, Noble, & Lee, 2016):

4.2.1 Xiaomi Mi Band 4

De todas, esta é provavelmente a mais acessível e a mais conhecida, o que se pode dever ao crescimento da Xiaomi no mercado da tecnologia móvel.

Esta banda ou pulseira possui um design bastante ergonómico, com uma durabilidade de bateria até 30 dias, à prova de água e poeiras. Possui também Bluetooth para ligação ao *smartphone* Android ou iOS, tem funcionalidades de desbloqueamento do telemóvel, mede distâncias percorridas, calcula calorias perdidas, monitoriza o sono, os batimentos cardíacos, os passos, etc.

Pelo preço, esta pulseira tem uma autonomia impressionante e tem funcionalidades interessantes para quem pretende realizar um rastreio da sua performance de treino (Mi, s.d.).

4.2.2 Fitbit Charge HR

A um preço um pouco superior que o Xiaomi Mi Band 4, este *wearable* tem como característica principal o medidor de batimentos cardíacos, uma vez que possui um pequeno ecrã onde mostra os mesmos em tempo real. Esta banda também possui um analisador de sono que mede os batimentos cardíacos durante todo o ciclo de forma a perceber quando o utilizador está, efetivamente, a dormir.

Tem funcionalidades de contagem de passos, notificações e customizações que podem ser ajustadas através de uma aplicação específica desta banda (Fitbit, 2018).

Contudo, baseado no artigo de Sarah Stahl (Stahl, An, Dinkel, Noble, & Lee, 2016), onde é discutida a precisão do monitor de batimentos cardíacos dos *smartwatches* e *fitness bands*, o Fitbit Charge HR tem um maior desvio médio do batimento cardíaco do que a concorrência. Isto comprova que a sua precisão e fiabilidade são menor do que os restantes, o que iria prejudicar a solução descrita nesta dissertação, uma vez que se procura a maior precisão de dados possível.

4.2.3 Samsung Gear Fit 2

Um pouco à parte do Samsung Gear S3, esta banda pretende-se focar na monitorização do exercício físico do indivíduo, retendo algumas funcionalidades de um *smartwatch*.

A novidade desta versão é a inclusão de GPS que monitoriza o posicionamento do utilizador e calcula a distância percorrida. Tal como a Xiaomi Mi Band, monitoriza também o batimento cardíaco e analisa o ciclo de sono.

Excepcionalmente, esta banda monitoriza uma vasta gama de exercícios, desde treinos de natação até a um simples agachamento, o que a torna uma opção mais vantajosa em relação ao Mi Band.

Como ponto negativo, esta banda não tem suporte para iOS, ou seja, só é possível ser utilizada por aqueles que têm um *smartphone* Android (Samsung, Samsung Gear Fit 2, s.d.).

É de notar que é possível consultar as mensagens e chamadas recebidas no *smartphone* quando emparelhado com esta banda. Com uma autonomia de dois/três dias, a Techradar (Faulkner, 2018) refere que é uma versão muito bem conseguida em termos de design e de conforto e que, pelo preço, é considerada uma excelente peça de tecnologia para quem quer começar com uma *fitness band*.

Esta *fitness band* não integra funcionalidades comuns como o alarme, levando a Techradar a recomendar o modelo Pro (Samsung, Gear Fit2 Pro, 2018) que conta com um ecrã AMOLED e funcionalidades para a natação, torna-o superior à Gear Fit 2 (Faulkner, 2018).

4.2.4 Sony SmartBand

Dos exemplos dados, esta é a banda que mais tenta inovar com o conceito. Possui funcionalidades semelhantes a todas as outras bandas, como monitorização do sono, dos batimentos cardíacos, da distância percorrida e das calorias perdidas. No entanto, monitoriza tudo o que está a acontecer com a pessoa, como, por exemplo, onde anda, o número de fotos tiradas, o tempo gasto a ouvir música, nas redes sociais, na internet, a jogar e a ler livros.

Esta banda é resistente a água e tem um software denominado de *Lifelog* que regista todo o tipo de informação referida em cima. Posteriormente, essa informação é enviada através de Bluetooth para o *smartphone*, uma vez que esta banda, sendo apenas uma bracelete de borracha, não possui qualquer tipo de visor.

Como aspeto negativo, esta banda não tem versão para iOS, estando disponível somente para Android, o que diminui a sua competição no mercado.

Por último, é de referir ainda que tem uma autonomia de 5 dias, dependendo do uso da sua funcionalidade de vibração (Trew, 2014).

4.3 Aplicações de Auxílio ao exercício físico (Fitness)

Como referido anteriormente, nota-se um crescente uso e desenvolvimento de aplicações de *fitness* com o objetivo de auxiliar o indivíduo durante o exercício, diminuindo a sua necessidade de depender dinheiro num ginásio ou PT.

Estas aplicações são, maioritariamente, gratuitas, podendo também ser pagas, como é o caso dos pacotes *premium* que contam com mais exercícios e com um maior apoio ao utilizador.

De seguida, serão enunciados alguns exemplos das aplicações mais utilizadas nos *smartphones*, segundo a Ekonomista (Duarte, 2017).

4.3.1 Zova

Zova é uma aplicação móvel disponível apenas para dispositivos iOS, com o objetivo de auxiliar o treino especializado para mulheres. Disponibiliza mais de 200 treinos desenvolvidos por profissionais na área que vão sendo ajustados mediante o nível de experiência do utilizador e a evolução do mesmo.

A aplicação apresenta-se com um sistema de integração com Apple Watch que monitoriza todos os exercícios feitos. Funciona também em sintonia com a aplicação nativa iOS Saúde, inserindo e lendo informações da mesma para obter o maior número de dados possível, de forma a proporcionar uma melhor experiência ao utilizador.

Conforme esperado, a aplicação envia mensagens de motivação para o dispositivo, mantendo o utilizador focado.

Foi desenvolvida também uma funcionalidade de “rede social” onde os utilizadores podem comparar resultados obtidos, mantendo-os informados dos resultados da comunidade e motivando o seu espírito de “competição”.

É ainda importante referir que todos os exercícios possuem um vídeo associado, explicando o posicionamento e a sua forma de execução.

Por último, a fonte de receita da aplicação são os pacotes *premium* que permitem um maior número de treinos personalizados, consoante a zona do corpo pretendida (Apple, Zova, 2017).

4.3.2 Nike + Training Club

É talvez a aplicação móvel de *fitness* mais utilizada no mercado, não só por pertencer a uma das maiores marcas de desporto (Finance, 2019), mas também por exibir grandes quantidades de publicidade, onde, normalmente, estão envolvidas celebridades.

Esta aplicação está presente nos dispositivos Android e iOS, não contando com integração com *smartwatches*. O número de exercícios disponíveis não passa dos 160, sendo que estes, tal como na Zova, estão categorizados por estilo de treino e nível de experiência.

Normalmente, uma aplicação deste tipo tem alguma forma de monetização que passa pela inclusão de publicidade ou de pacotes com mais exercícios e funcionalidades. No entanto, a Nike + Training Club foge à regra neste aspeto, sendo completamente grátis.

Possui ainda a particularidade de ter uma funcionalidade de treino rápido, sem utilização de equipamento extra, focado nos utilizadores que têm pouco tempo para ir ao ginásio ou que pretendam realizar treinos mais rápidos em casa.

Apresenta também um vídeo especializado para cada exercício, auxiliando o indivíduo aquando da execução do mesmo.

Por último, toda a informação é gravada na aplicação, de forma a verificar a performance do utilizador, assim como na aplicação nativa iOS Saúde, com o objetivo de centralizar toda a informação (Apple, Nike Training Club, 2017).

4.3.3 Runkeeper

Esta aplicação está disponível em Android e iOS e tem integração com *smartwatches* como Apple Watch.

A Runkeeper é a aplicação com mais tempo de mercado e é a mais especializada em corridas, passeios de bicicleta e até descidas de Ski.

Como componente principal, a aplicação utiliza o GPS para fazer todos os cálculos de distância. Tem ainda, como característica única, a disponibilização em tempo real via áudio de dicas para o treino. Tem uma funcionalidade de integração com a música disponível no seu telemóvel, permite partilhar fotos durante uma corrida e comparar resultados e definir metas pessoais. Quando são alcançadas novas pontuações, a aplicação alerta o utilizador e permite partilhar as mesmas, de forma a fomentar a competitividade. Com este conceito de “jogo”, a aplicação torna-se bastante única e popular.

A Runkeeper também oferece uma versão Elite (*premium*) que permite ocultar as publicidades, comparar resultados anteriores, orientar metas automaticamente definidas pela aplicação e ter um acompanhamento em tempo real.

Tal como as outras aplicações acima descritas, esta também usa a aplicação Saúde para ler e escrever informação, de uma forma a recolher o maior número de dados na primeira utilização (Apple, Runkeeper, 2017).

4.4 Comparações finais

Após realizada a descrição de vários exemplos de *smartwatches*, *fitness bands* e aplicações móveis, será feita uma comparação dos modelos referidos de forma mais visual, com o objetivo de determinar as funcionalidades que poderão gerar valor à solução proposta nesta dissertação.

É importante referir que o foco será a obtenção de dados a partir dos *wearables* fornecidos à aplicação móvel para dar recursos ao algoritmo de geração de treinos.

4.4.1 Smartwatches

Os pontos apresentados na Tabela 2 são os mais relevantes para o tema desta dissertação. Nas secções acima apresentou-se uma descrição mais detalhada de cada um deles.

Tabela 2 - Comparação entre *Smartwatches*

	Sensor de batimentos cardíacos	Integração com aplicações de exercício físico	Aplicação nativa para consulta de dados	Customização
Apple Watch	X	X	X	X
Samsung Gear S3	X	X	X	X
TAG Heuer Connected Modular	-	X	X (*)	X

* - Apesar do TAG Heuer ter várias aplicações para consulta de dados meteorológicos, mensagens, etc, não tem consulta nativa de informações sobre exercício físico, pelo que se terá de optar por utilizar aplicações de terceiros para verificar essa informação.

Com base na tabela acima referida, o Samsung Gear S3 e o Apple Watch são os mais completos, tendo também em conta que têm um enorme leque de funcionalidades, como a consulta de mensagens, meteorologia, etc. Além disso, possuem um preço bastante mais acessível relativamente ao TAG Heuer.

O TAG Heuer opta por uma abordagem semelhante aos relógios tradicionais, tendo a possibilidade de instalação de aplicações como Uber ou Spotify. Em contrapartida, não integra sensores de batimentos cardíacos que seriam necessários para o objetivo da solução.

4.4.2 Fitness bands

Tal como na secção 4.4.1, a comparação das várias *fitness bands* será feita de acordo com os aspetos mais importantes para o tema desta dissertação, uma vez estes terão maior foco no desenvolvimento da solução proposta.

Tabela 3 - Comparação entre *Fitness bands*

	Sensor de batimentos cardíacos	Integração com aplicações de exercício físico	Aplicação nativa para consulta de dados	Customização
Xiaomi Mi Band 4	X	X	X	X
Fibit Charge HR	X	X	X	X
Samsung Gear Fit 2	X	X	X	X (**)
Sony SmartBand	- (*)	X	X	X (**)

* - A versão descrita nesta dissertação não possui sensor de batimentos cardíacos, no entanto, a nova versão (Sony SmartBand 2) já integra esta funcionalidade (Sony, 2018).

** - Possuem apenas customização a nível da cor da bracelete. A Xiaomi Mi Band 4 tem braceletes de várias cores e materiais, dando uma maior possibilidade de customização.

Perante o exposto na Tabela 3, estas *smartbands* são equivalentes entre si, uma vez que existe um *standard* de funcionalidades que cada uma deve possuir para ser competitiva.

Em relação à Mi Band 4, esta possui uma interatividade bastante reduzida comparativamente com a concorrência. Apesar disso, é a *fitness band* mais personalizável e acessível.

Em termos de funcionalidades inovadoras, é o Sony SmartBand que tenta criar algo diferente através da ferramenta *Lifelog* que regista toda a atividade do utilizador, de acordo com o referido na secção 4.2.4. Por outro lado, não possui sensor de batimentos cardíacos, o que é fulcral nos dias de hoje em qualquer *fitness band*. Apesar disso, e tendo em consideração que a segunda versão já possui este sensor, o Sony SmartBand é a única que aposta na inovação deste conceito.

De salientar ainda que a precisão da medição dos batimentos cardíacos é importantíssima para a solução final. Uma vez que é nesta variável que a criação dinâmica de exercícios é baseada, será utilizado o melhor *wearable* disponível no mercado para a recolha desta informação.

Perante o contexto referido e as comparações acima realizadas, é notória a superioridade do Apple Watch em relação ao monitor de batimentos cardíacos. De ter ainda em consideração todo o seu ecossistema com o iPhone (aplicação Saúde) e o facto de ser um dispositivo em constante evolução.

4.4.3 Aplicações de Auxílio ao exercício físico

Como o objetivo da solução é desenvolver uma aplicação móvel, o estado de arte dará maior ênfase a este ponto, uma vez que a aplicação terá de possuir algo que a faça distinguir do que existe atualmente no mercado.

Apesar de ser uma aplicação destinada ao PT, algumas das funcionalidades abaixo referidas poderão estar presentes de forma a tornar o produto final mais completo. Apesar disso, terá como principal valor o algoritmo de criação de exercícios baseado nos batimentos cardíacos e variáveis como a parte de corpo que pretende exercitar e o objetivo a atingir pelo utilizador.

Tabela 4 - Comparação entre aplicações móveis de auxílio ao treino

	Tutorial de execução dos exercícios	Integração com a Health App	Sistema de Notificações	Comparação com resultados de outros utilizadores	Pacote <i>Premium</i>	Histórico de exercícios	Integração com <i>Wearable</i>
Zova	X	X	X	X	X	X	X
Nike + Training Club	X	X	X	-	-	X	-
Runkeeper	X	X	X	X	X	X (*)	X

* - É necessário obter a versão paga.

Com base na análise feita, é possível perceber o padrão das aplicações móveis mais usadas, tem em conta que estas tentam obter a atenção dos utilizadores através de publicidade e de campanhas de *marketing*.

Existe ainda necessidade de estabelecer um grande número de exercícios estudados por profissionais na área do *fitness* e exercício físico que integram as funcionalidades de histórico de resultados e tutoriais de execução de treinos complementando a aplicação Saúde da Apple.

Posto isto, existem algumas funcionalidades específicas para cada aplicação que se encontram limitadas pela sua versão *Premium/Elite*.

Começando pela Nike + Training Club que, apesar de ter a vantagem de estar relacionada com uma marca de desporto, peca por não possuir integração direta com o Apple Watch ou outro *wearable*. Da mesma forma, não dá uso ao potencial de se tornar numa rede social de competição, uma vez que já existe uma outra aplicação móvel da Nike (Nike + Run Club) que é especializada neste conceito (Apple, Nike+ Run Club, 2018).

Por sua vez, a Zova que, segundo o iTunes, é aplicação com mais prémios da Apple Design e a melhor aplicação da App Store e Editor's Choice (Apple, Zova, 2017), dá uso a todo o tipo de funcionalidades, tirando partido ainda do Apple Watch e do conceito de rede social, onde poderá proporcionar uma melhor experiência para os utilizadores aficionados no exercício físico.

O mesmo se pode aplicar ao Runkeeper, que se tenta destacar de marcas já estabelecidas através de funcionalidades como integração do Spotify e de várias aplicações de centralização de informação como a aplicação Saúde ou Fitbit. Inclui também o conceito de rede social, com o acréscimo de possuir funcionalidades de convite para praticar desporto de forma a, posteriormente, comparar resultados entre indivíduos.

Seria interessante desenvolver algumas destas funcionalidades como o sistema de notificações, o histórico de resultados, a integração com o Apple Watch e a integração com aplicações nativas como a aplicação Saúde. Isto, em conjunto com o algoritmo proposto neste documento, traria algo inovador que facilitaria o acompanhamento dos clientes por parte do PT.

Inserir página em branco apenas se necessário de modo a que o próximo capítulo comece numa página à direita

5 Análise

Neste capítulo, irá ser feita uma análise mais profunda à solução, com vista em detalhar como esta estará estruturada tecnicamente, segundo as boas práticas, padrões e regras da programação aprendidos.

Esta alínea subdivide-se nos seguintes pontos: design da solução para o problema; design arquitetural; fluxograma; design da interface gráfica; criação de planos adaptados e dinâmicos; soluções/abordagens existentes e suas avaliações; programas e equipamentos necessários e, por último, tecnologias utilizadas.

5.1 Design da solução

A abordagem para o design da solução foi concebida em função das funcionalidades pretendidas para o projeto, com a preocupação de tornar o projeto escalável e totalmente modular, delegando as responsabilidades a entidades independentes.

Foi definida a seguinte prioridade quanto aos novos módulos e funcionalidades:

- Desenvolver a base para a comunicação entre o iPhone e o Apple Watch;
- Desenvolver a base para a comunicação com o HealthKit para obtenção da informação dos sensores e respetiva escrita na aplicação de Saúde;
- Criar e alterar a API e a base de dados para estarem preparadas para as novas funcionalidades e informações;
- Desenvolver as novas funcionalidades de gestão de informação dos clientes do PT:
 - Criação de gráficos de histórico das informações;
 - Desenvolver algoritmo de geração automático de treino;
 - Sistema de notificações sobre o ritmo cardíaco do cliente para o PT;
 - Aplicação para o Apple Watch para o cliente visualizar os resultados obtidos;
- Melhorias na base da aplicação já estabelecida.

Primeiramente, houve a possibilidade de desenvolver o projeto em **Swift**, uma vez que é uma nova linguagem que facilita o desenvolvimento. No entanto, existe uma maior quantidade de material de apoio disponível sobre o **Objective-C** e, tendo ainda em conta que a maior *expertise*

na ITSector é em relação ao Objective-C, decidiu-se utilizar esta linguagem. Posteriormente, será feita uma referência mais profunda a estas duas linguagens de programação no capítulo **Erro! A origem da referência não foi encontrada..**

De seguida, foi decidido construir uma aplicação acessível para o iPhone, sem funcionalidades muito complexas, **de forma a facilitar a leitura e manipulação da informação**. O mesmo acontece para a aplicação do Apple Watch, uma vez que esta iria ser uma pequena extensão à solução e uma forma de oferecer alguma informação aos clientes do PT.

Posto isto, foi definido o Modelo de Domínio, que ilustra as componentes principais do negócio das **aplicações do iPhone e Apple Watch**, sem dar referência à estrutura do chamado *Middleware*.

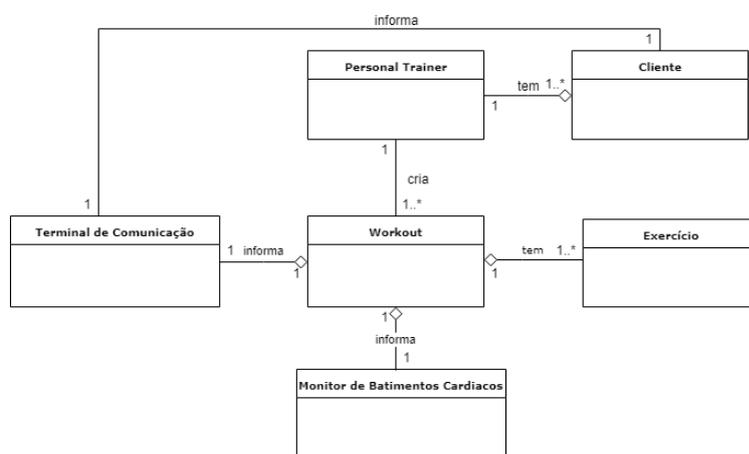


Figura 4 - Modelo de Domínio

Com base na Figura 4, é possível observar que o **“Workout”** é a componente mais importante do negócio, sendo através dela que se pode gerar valor para o **“Personal Trainer”** e para o **“Cliente”**.

O **“Monitor de Batimentos Cardíacos”** é também relevante, uma vez que é este que terá acesso aos batimentos cardíacos de forma a informar o **“Workout”** que existe um novo batimento. Este, por sua vez, informa o **“Terminal de Comunicação”** que está diretamente ligado ao **“Cliente”**.

Uma possível implementação com o *Middleware* seria através de uma ligação entre o **“Personal Trainer”** e a API que exponha a lógica do negócio, uma vez que é este que tem acesso aos treinos de cada cliente.

A lógica do negócio estaria centralizada no *Middleware*, uma vez que os *smartphones* podem não apresentar grande capacidade de memória para esta lógica, o que poderá **deteriorar a experiência do utilizador**. Assim, o *smartphone* terá apenas a responsabilidade de mostrar a informação, controlar as ações do utilizador e manipular os treinos através do **HealthKit**, comunicando com o *smartwatch* para este obter a informação necessária.

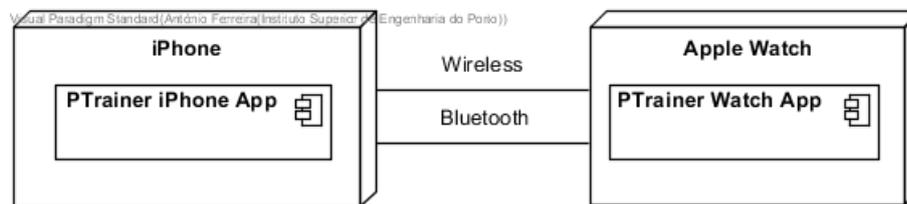


Figura 5 - Diagrama de Implantação

O projeto será então constituído por uma **aplicação para o iPhone** direcionada para o personal trainer e uma **aplicação para o Apple Watch** direcionada para o cliente. Estes dispositivos comunicam através de Bluetooth e Wireless como se pode verificar na Figura 5. Poderia ainda haver mais dois componentes que seriam a **API** e a **base de dados** do *Middleware*.

A aplicação para o iPhone obteria a informação do *Middleware* que não é nada mais que uma API (ponte) entre a base de dados e a aplicação móvel. À entrada da aplicação **seriam invocados todos os serviços para popular os ecrãs**, uma abordagem que, caso decorresse numa aplicação de maior escala, poderia ser algo falível em termos de performance e usabilidade. No entanto, como se trata de uma aplicação de pequena escala, torna a experiência de usabilidade mais rica e fluída.

Uma aplicação para o Apple Watch contem dois pacotes nominados de **Watch App Bundle** e **WatchKit Extension Bundle**. O primeiro pacote contém todas as *views* e *storyboards* da aplicação, enquanto que o segundo inclui todo o código da aplicação. Esta interação entre os pacotes e o Apple Watch e o iPhone será detalhada de uma forma mais visual na Figura 6 (Apple, The Watch App Architecture, 2016).

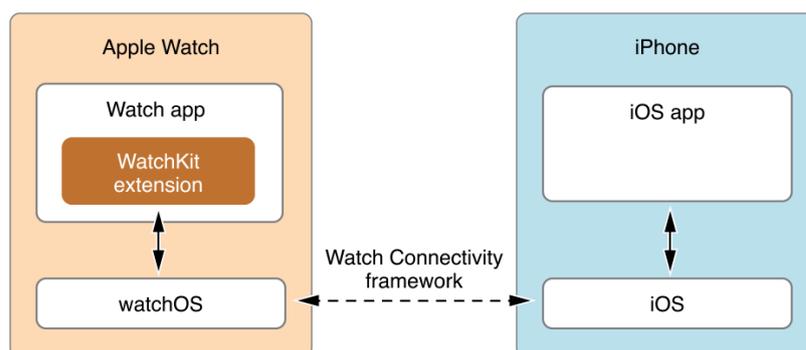


Figura 6 - Diagrama referente à interligação do Apple Watch e do iPhone (Gusgård, 14)

A ligação é feita através de uma *framework* oficial da Apple, chamada Watch Connectivity (Apple, Watch Connectivity, 2018), onde se pode observar que se trata de um **“gestor”** de ligações de iPhone para Apple Watch e vice-versa, uma vez que não há outra forma de enviar mensagens entre Watch e iPhone sem usar esta componente. Tendo em conta que esta é oficial,

permanece atualizada e otimizada e permite o uso dos sensores referidos em cima para enviar mensagens ou ficheiros entre os dispositivos.

A comunicação com o HealthKit é da responsabilidade a aplicação do Apple Watch. Uma vez que esta tem acesso aos sensores, contará com uma classe que será responsável por fazer esta ligação e fornecer a informação que será pedida. Assim, o Apple Watch enviará apenas a informação para o iPhone, sendo da responsabilidade deste enviar essa informação para a API que será depois inserida na base de dados.

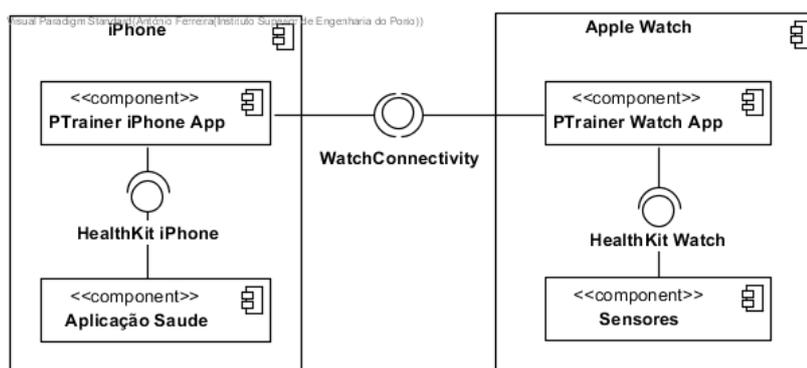


Figura 7 - Diagrama de Componentes

É importante referir que a aplicação do iPhone também terá de comunicar com o HealthKit para armazenar os resultados na aplicação de Saúde. Na Figura 7, são ilustrados os componentes da solução, ou seja, como o Apple Watch e o iPhone comunicam entre si e como estes comunicam com o HealthKit. Também aqui poderia ser adicionada mais uma componente que representaria o Middleware com uma ligação **HTTPS** com o iPhone.

Parte do foco deste projeto seria acoplar todas estas entidades e criar uma plataforma interligada, mas, devido ao facto de ter havido **mudança da entidade empregadora**, foi impossibilitado o desenvolvimento sobre o *Middleware* e a base de dados. Desta forma, houve um maior foco no desenvolvimento da aplicação móvel, uma vez que esta é um módulo isolado da aplicação principal PTTrainer.

5.1.1 Requisitos Funcionais

Com o início do desenvolvimento do design da aplicação móvel, foram definidos os requisitos funcionais da mesma. Estes representam as funcionalidades chave e serão distribuídos pelos demais Casos de Uso ou *Use Case* (UC).

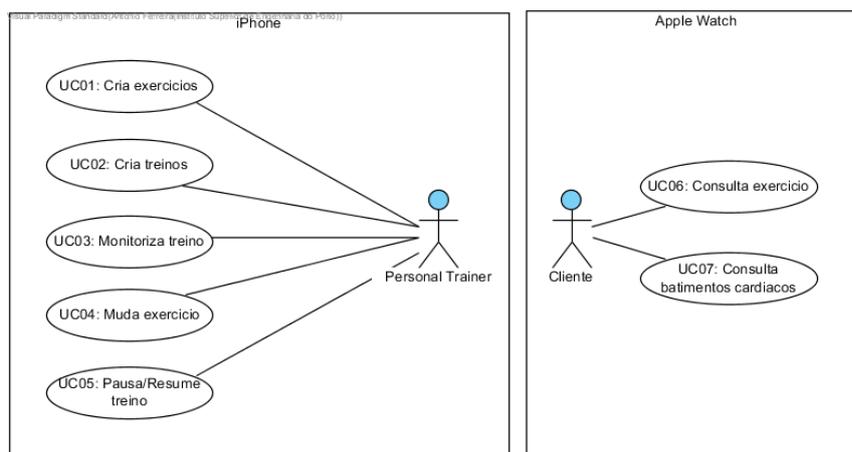


Figura 8 - Diagrama de Casos de Uso

De seguida, serão enunciados os UC's de maior destaque, conforme o ilustrado na Figura 8:

- ✓ **Caso de Uso 1:** O PT pretende criar exercícios para poder adicionar aos seus treinos;
- ✓ **Caso de Uso 2:** Com a componente de construção de exercícios criada, é expectável que o PT possa elaborar vários treinos compostos por exercícios e intensidade;
- ✓ **Caso de Uso 3:** O PT pretende monitorizar os treinos dos seus clientes em tempo real. Para isso, é necessário desenvolver uma interface de gestão de informação com a aplicação nativa iOS Saúde, através do HealthKit e com o Apple Watch e iPhone, assim como a estrutura de comunicação da aplicação móvel com o Apple Watch;
- ✓ **Caso de Uso 4:** O PT pretende mudar de exercício sempre que desejar. Usando a estrutura de comunicação entre o iPhone e o Apple Watch é possível enviar mensagens para o Apple Watch para este alterar o exercício em execução;
- ✓ **Caso de Uso 5:** O PT pretende pausar e resumir um treino, uma vez que o cliente poderá estar cansado e o seu batimento cardíaco poderá exceder o limite definido pelo treino;
- ✓ **Caso de Uso 6:** O Cliente pretende consultar o exercício em execução através do Apple Watch. Para isso será necessário criar uma interface gráfica para a aplicação;
- ✓ **Caso de Uso 7:** O cliente pretende consultar o seu batimento cardíaco em tempo real através da aplicação do Apple Watch, usando a mesma interface gráfica criada no Caso de Uso 6.

5.1.2 Requisitos Não Funcionais

Quanto aos requisitos não funcionais, iram de encontro ao Desempenho, Compatibilidade, Usabilidade, Robustez e Escalabilidade da aplicação móvel, não só para satisfazer as partes

interessadas, mas também para relacionar o que é pretendido da Engenharia Informática. É necessário distribuir responsabilidades, modular e baixar o acoplamento de uma solução para mais facilmente se efetuar uma manutenção caso haja novos desenvolvimentos, sem interferir com a solução atual.

- **Desempenho** – o utilizador final pretende que a aplicação móvel seja responsiva e que não haja quebras na fluidez para desfrutar da melhor experiência possível;
- **Compatibilidade** – apesar de a solução abranger apenas o Apple Watch, é necessário que esta esteja pronta para ser executada com um outro *wearable*;
- **Usabilidade** – como foi referido, pode haver muitos clientes que não estejam à vontade com aplicações móveis, sendo necessário garantir a melhor usabilidade possível para evitar que o utilizador passe demasiado tempo a tentar executar uma funcionalidade;
- **Robustez** – a solução terá de ser o mais robusta possível, sem falhas ou incongruências, com os devidos avisos de uma errada execução para o utilizador;
- **Escalabilidade** – é necessário que a aplicação móvel esteja pronta para receber novas funcionalidades que facilmente sejam integradas no projeto, sem a necessidade de trabalho “extra”.

5.2 Design Arquitetural

O design arquitetural de uma solução é de extrema importância, uma vez que define a sua estrutura e como esta irá executar as funcionalidades pretendidas, detalhando os padrões utilizados. Logo, nesta secção, será feita uma análise extensiva aos padrões utilizados e a algumas alternativas existentes.

5.2.1 Model View Controller (MVC)

Programar usando este padrão arquitetural faz-se através de três componentes, onde objetos de diferentes classes podem tomar conta do domínio da aplicação (**Model**), o estado gráfico da aplicação (**View**) e a interação do utilizador com a View e o Model (**Controller**) (Krasner & Pope, 1988).

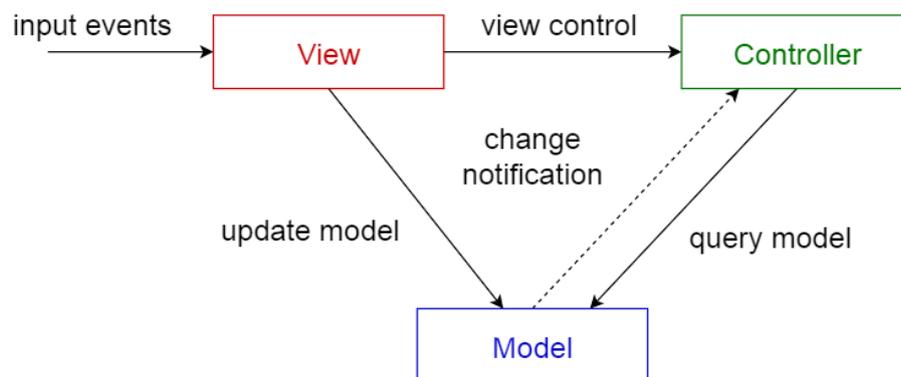


Figura 9 - Padrão MVC

Com base na Figura 9 (Mallawaarachchi, 2017), é perceptível a interação entre estes três componentes, sendo a sua simplicidade de abordagem e facilidade de teste e manutenção que o tornam um dos padrões mais conhecidos.

Seguidamente, serão explicados com maior detalhe cada componente deste padrão:

- ✓ **Model ou Modelo** – é específico do domínio e normalmente representa uma entidade do mesmo. Agrega informação e é ele que dita o que deve ou não possuir. Usualmente, os dados do Modelo são apresentados na View e são manipulados pelo Controller, consoante a interação do utilizador na View;
- ✓ **View ou Vista** – lida com tudo que é gráfico, pede informação do Modelo, apresentando-a;
- ✓ **Controller ou Controlador** – filtra toda a interação com a View e delega o que pode ou não acontecer. Esta delegação permite depois manipular a informação de um Modelo que esta a ser manipulado pela View.

Este padrão arquitetural é bastante utilizado em aplicações móveis e foi o escolhido para o desenvolvimento desta solução devido às bases académicas e profissionais obtidas. Este padrão é bastante interessante também pelo facto de ser desenhado para projetos mais pequenos e de rápido desenvolvimento (Wayner, 2018).

5.2.2 Model View ViewModel (MVVM)

Considerado uma modificação do padrão MVC, este padrão é também composto por três componentes: o Model, a View e o ViewModel. Neste caso, em vez do Controller, existe o ViewModel, que tem como responsabilidade converter os objetos do Model de forma que possam ser facilmente geridos e apresentados. No entanto, a grande diferença está na separação da lógica de negócio e de programação da interface gráfica. Ou seja, facilmente um

designer pode trabalhar independentemente de um programador e vice-versa (Microsoft, The MVVM Pattern, 2012).

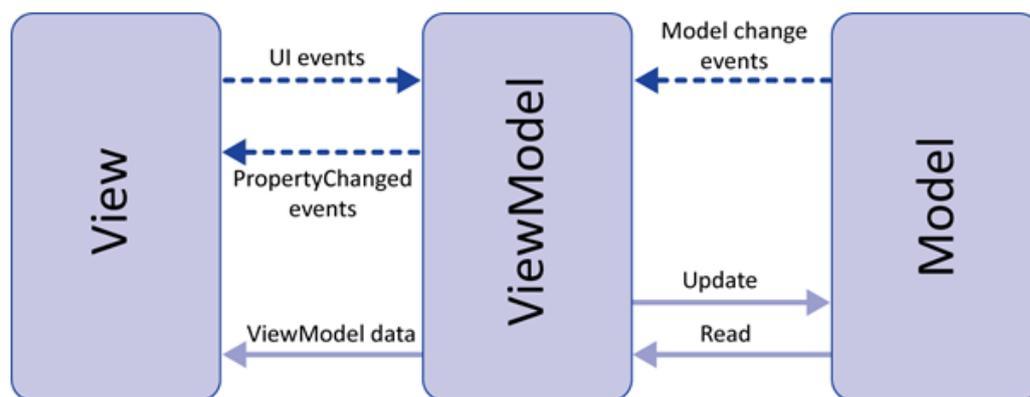


Figura 10 - Padrão MVVM

De certa forma, o ViewModel é mais um Model do que uma View e é bastante semelhante ao Controller, uma vez que trata da lógica da View. Apesar disso, o seu menor acoplamento torna este padrão mais complexo, uma vez que separa as responsabilidades e desacopla as entidades (Figura 10 (Microsoft, Implementing the Model-View-ViewModel Pattern, 2017)).

- ✓ **Model ou Modelo** – tal como no MVC, é específico do domínio e normalmente representa entidades do negócio. Neste caso, esta sempre à espera de receber modificações e de informar o ViewModel;
- ✓ **View ou Vista** – mais desacoplado da lógica de negócio, a View só terá de mostrar a informação dada pelo ViewModel e de notificar todas as interações ao mesmo;
- ✓ **ViewModel** – como referido, este está mais acoplado ao Model e menos à View e tem a característica de construir a informação do Model de forma que a View a interprete sem a necessidade de a enviar o Model.

No caso desta solução, não foi adotada esta arquitetura, uma vez que a base já estava construída com o padrão do MVC. Apesar disso, esta abordagem seria bastante interessante, uma vez que separa as responsabilidades e facilita a manutenção.

5.3 Fluxograma

Antes do desenvolvimento da aplicação, foi definida uma abordagem de reutilização do fluxo nativo de navegação da Apple, onde existe uma view base e todas as outras iriam surgir a partir desta, criando assim uma pilha de views de forma hierárquica. Posto isto, foi **então definida a view principal da aplicação** e toda a sua hierarquia, tendo em conta os aspetos mais fulcrais para as partes interessadas (PT).

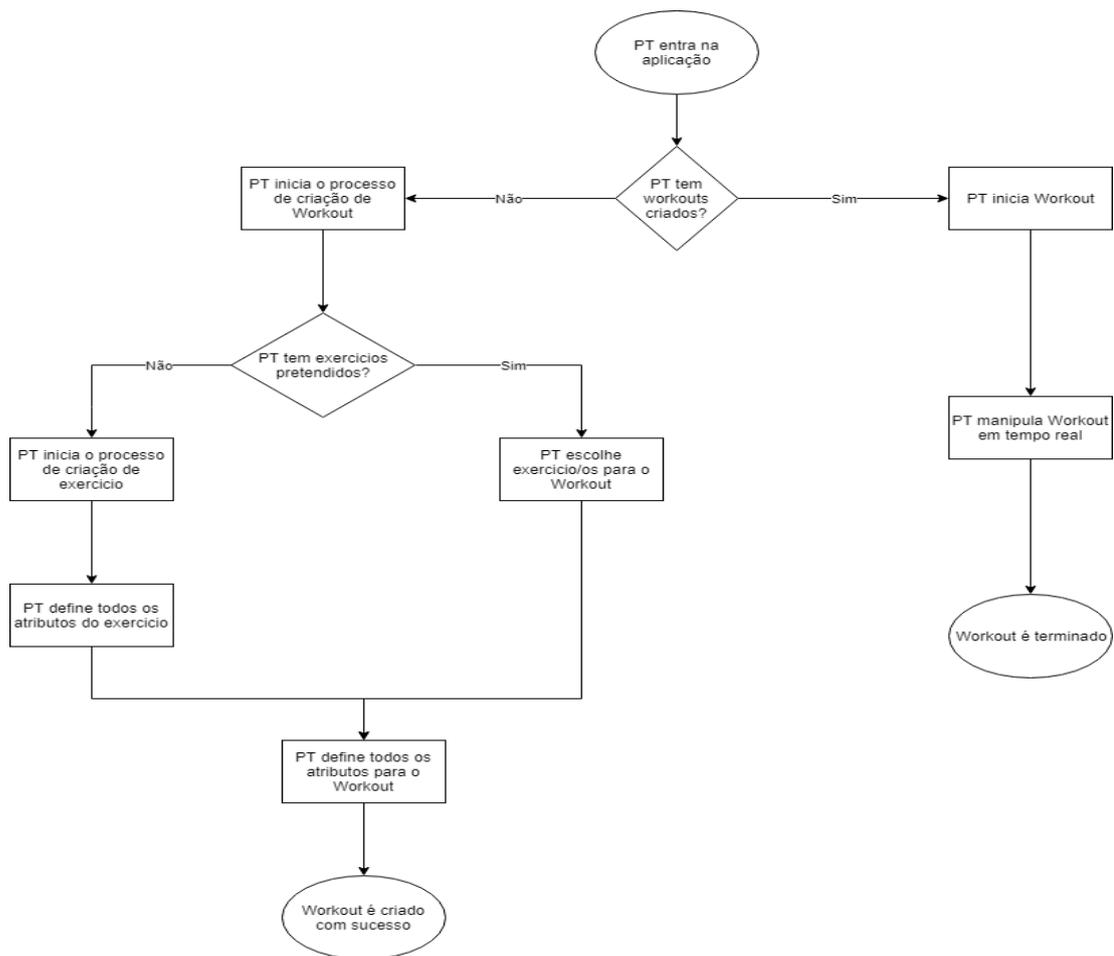


Figura 11 - Fluxograma da aplicação

Perante a Figura 11, é perceptível que esta lógica pode ser colocada em qualquer aplicação como se de um novo módulo se tratasse, uma vez que toda a sua lógica depende apenas da primeira view, o que ajuda na sua modularização.

5.4 Design da Interface Gráfica

Desde a conceção da ideia para o documento, o foco seria utilizar todas as tecnologias nativas e mais recentes para não comprometer futuras manutenções e tornar o produto final facilmente escalável.

O mesmo foi pensado para as interfaces gráficas. A ideia seria não alienar os utilizadores do iPhone. Para isso, esta teria de seguir o guia de desenvolvimento de interfaces gráficas da Apple (Apple, Human Interface Guidelines, 2018), que define espaçamentos, seleção de cores, padrões gráficos, ícones e imagens, entre outros. Ou seja, não existiria qualquer tipo de menu que, muitas vezes, é visto nas aplicações de Android. Assim, **todo o fluxo será feito através de ícones e botões que estão dispostos conforme o contexto da view.**

Como se pode observar nas figuras abaixo, a cor principal utilizada é o **azul**. Esta permite identificar a marca da aplicação, mantendo conciso o objetivo da view e apresentando sempre um título que permite identificar o propósito da mesma.

Sempre que for necessário, serão utilizados **ícones nativos e simples** que facilmente identificam o seu objetivo. Na Figura 14, é possível observar a utilização de elementos gráficos de texto e de seleção de opções nativas da Apple, com algumas personalizações de tipo de letra e de cor. O mesmo se pode observar na Figura 16, onde são **reutilizados os mesmos controlos gráficos** para inserção da informação do exercício a ser criado.

Já na Figura 18 é possível ver o único ecrã que estará presente na aplicação do Apple Watch e, conforme o guia da Apple, o background deve ser **preto** para destacar o seu conteúdo, uma vez que se trata **de um dispositivo bastante pequeno**. Por sua vez, a cor **vermelha** mostra os batimentos cardíacos e o batimento máximo de forma a acentuar esta informação para o cliente. Mais uma vez, a interface gráfica foi realizada utilizando os componentes gráficos nativos do Apple Watch, conforme o que foi realizado na aplicação do iPhone.

É de notar que todas as figuras abaixo apresentadas são apenas um **esboço** do que é pretendido e, portanto, não são definitivas.

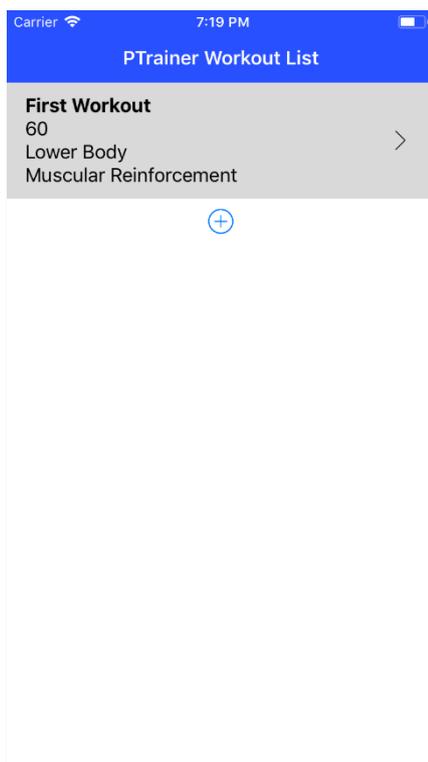


Figura 12 - Primeiro ecrã da aplicação

Na Figura 12, é possível observar a página principal da aplicação. É aqui que será apresentada a lista de treinos do PT, detalhando o **nome do mesmo, o tempo, a intensidade e o objetivo**. No final, encontra-se um botão para adicionar um novo treino, se assim for desejado.

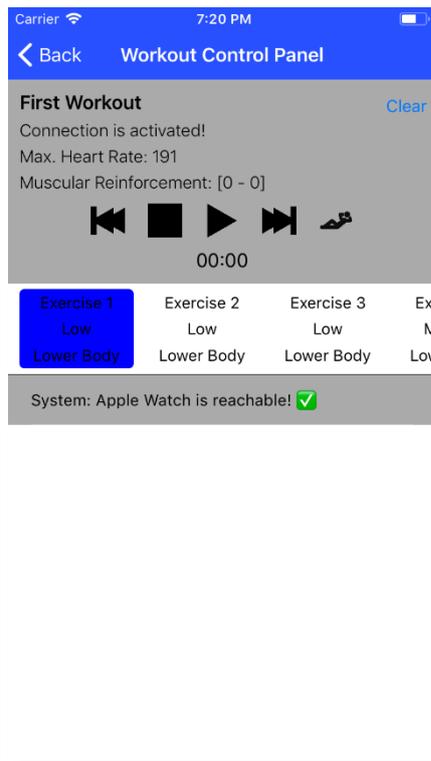


Figura 13 - Ecrã do Workout selecionado

Caso seja selecionado um treino, é apresentado o ecrã de monitorização do mesmo (Figura 13). É aqui que é possível iniciar, pausar ou até parar o treino. Mostra também todos os exercícios configurados para este treino, onde é **possível selecionar um à escolha, avançar ou retroceder**. É ainda possível definir um período de descanso que é **personalizável** pelo PT.

Com estas funcionalidades de controlo, vem depois a lista onde é mostrada toda a informação sobre o treino. Esta pode ser **informação relativa aos batimentos cardíacos, aos exercícios que foram executados ou informação mais tecnológica**, como, por exemplo, quando o iPhone não consegue comunicar com o Apple Watch devido à distância.

The screenshot shows a mobile application interface titled "Workout Creator". At the top, there is a blue header bar with a back arrow and the text "Workout Creator". Below the header, the form is organized into sections with light gray backgrounds. The first section is "Workout Name:" with a text input field containing "Fast workout...". The second section is "Workout Time:" with a text input field containing "45 minutes...". The third section is "Training Type:". The fourth section is "Exercise Body Area:". The fifth section is "Add Exercises:" with a right-pointing chevron icon. The sixth section is "HIIT?" with a toggle switch that is currently turned off. At the bottom of the form is a prominent blue button labeled "Create". The status bar at the top of the phone shows "Carrier", signal strength, Wi-Fi, and the time "7:21 PM".

Figura 14 - Ecrã de criação de um Workout

Após a seleção do botão de criação de um novo treino, representado na Figura 12, será apresentado o formulário de criação de um treino (Figura 14). Aqui é **possível definir o nome, o tempo, o tipo de treino, a área do corpo que se pretende treinar, se o treino permite HIIT e a lista de exercícios**. Esta página obriga o preenchimento de todos os campos, uma vez que são variáveis necessárias para o bom funcionamento do algoritmo.

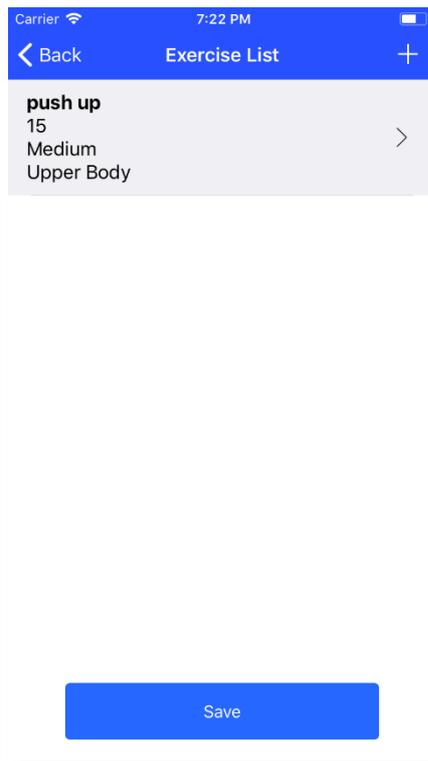


Figura 15 - Ecrã da lista de exercícios do Workout

Como referido na paragrafo anterior, quando é selecionada a opção de criar novos exercícios, é apresentado o ecrã da Figura 15. Aqui é apresentada **uma lista com os exercícios selecionados para o treino, mostrando toda a informação relativo aos mesmos**. Se for necessário criar mais exercícios, existe um botão no canto superior direito que direciona para o devido formulário. Caso todos os exercícios já estejam bem definidos, é só carregar no botão de guardar para associá-los ao treino.

Carrier 7:30 PM

Cancel Create Exercise Save

Exercise Name:
push ups

Exercise Duration:
15

Exercise Intensity:
Low
Medium
High

Exercise Body Area:
Lower Body Area
Upper Body Area
Core Body Area

Figura 16 - Ecrã de criação de exercícios para o Workout

Na Figura 16, é possível observar o formulário de criação de um exercício. É aqui que são definidos todos os atributos de **um exercício, nomeadamente a duração, o nome, a intensidade e a zona do corpo exercitada**. No canto superior direito da imagem existe um botão de guardar que, quando interagido, guarda o exercício na lista de exercícios do treino, surgindo posteriormente o ecrã da lista de exercícios.

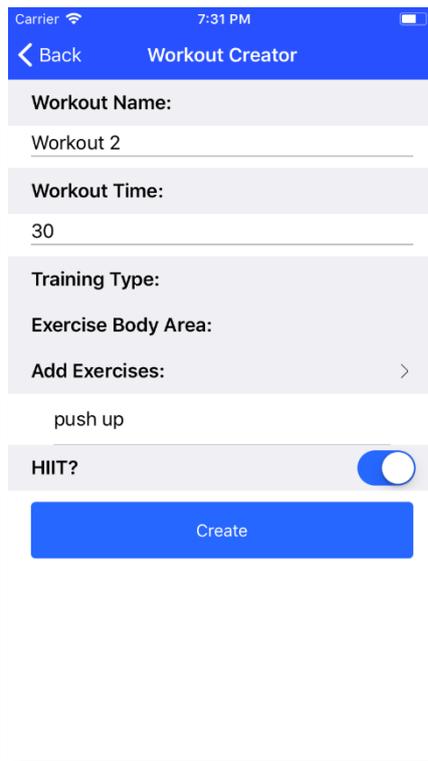


Figura 17 - Ecrã de criação do Workout preenchido

Quando a informação sobre o treino é preenchida, teremos um ecrã da Figura 17. Toda a informação introduzida é guardada através da interação com o botão criar, retrocedendo depois para o ecrã principal com a lista de treinos. Todo este fluxo é bastante simples, com o objetivo de ser o mais claro possível, possuindo cores facilmente destacáveis e dando prioridade à usabilidade da aplicação.

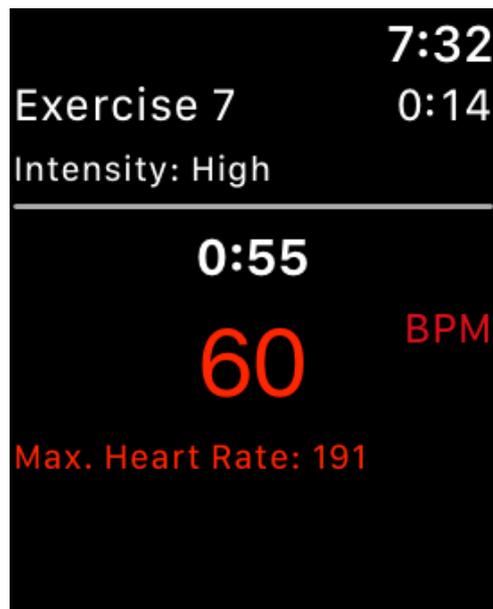


Figura 18 - Ecrã Principal da aplicação Apple Watch

Por último, na Figura 18 é apresentada a estrutura da aplicação Apple Watch. Esta contém apenas um ecrã, onde mostra toda informação necessária para o cliente do PT, ou seja, os **tempos de treino e de exercício atual, o batimento cardíaco e o batimento cardíaco máximo**. Esteve em discussão se faria sentido apresentar o batimento cardíaco ao utilizador ou ao PT, no entanto, definiu-se que o utilizador poderia observar essa informação, uma vez que lhe permite ter a noção de quando este está a chegar ao seu limite cardíaco.

5.5 Criação de planos adaptados e dinâmicos

Previamente à decisão dos tipos de treinos e suas intensidades, realizou-se uma pesquisa sobre regras de definição do batimento cardíaco máximo e intervalos de batimentos cardíacos de acordo com a intensidade, assim como margens de erros baseado no *wearable* utilizado.

Neste subcapítulo será explicado com maior foco estes aspetos e como os mesmos influenciaram o algoritmo proposto. Todos estes aspetos são baseados no artigo científico escrito por António Ferreira, Jorge Coelho e Nuno Nogueira (Ferreira, Coelho, & Nogueira, 2018).

5.5.1 Exercícios do Treino

Para esquematizar os exercícios, estes teriam de ser divididos por **intensidade/intensity**, ou seja, por facilidade de execução e por **área do corpo/body área**, ou seja, pela zona do corpo que se está a exercitar (Ferreira, Coelho, & Nogueira, 2018).

Com base nestas duas variáveis, seria mais fácil diferenciar os exercícios.

- **Intensidade** – baixa/low, média/medium e alta/high;

A intensidade num exercício é definida através de quão rápido/intenso é a prática do mesmo. Um fator que determina a intensidade é a frequência cardíaca, uma vez que esta tende a aumentar quando aumenta a intensidade do exercício (CardioSmart, 2015).

Por esta razão, a intensidade de um exercício está diretamente ligada a um intervalo de batimentos cardíacos, estando este aspeto detalhado com maior granularidade posteriormente.

- **Área do corpo** – inferior/lower, superior/upper e central/core;

Os exercícios podem exercitar várias partes do corpo (bíceps, tríceps, lombar, gêmeos, etc.) no entanto, para simplificar todas estas zonas, foram definidas três áreas, com o objetivo de facilitar a categorização de exercícios.

A partir deste conceito, é dada a Tabela 5 como exemplo:

Tabela 5 - Exemplo de uma lista de Exercícios e suas variáveis

Exercício	Área do corpo	Nível de Intensidade
Squat	Inferior	Baixa
Squat Jump	Inferior	Média
Tuck Jump	Inferior	Alta
Push Up Knees Down	Superior	Baixa
Push Up	Superior	Média
Burpee	Superior	Alta
Mountain Climber	Central	Baixa
Sit Up	Central	Média
Hollow Rock	Central	Alta

Muitos exercícios podem exercitar mais do que uma zona do corpo, no entanto, será referida no produto final a zona que é mais treinada.

A especificação técnica de cada exercício não será do interesse do projeto em questão, uma vez que será o PT a criar os seus exercícios, explicando a sua execução aos clientes. Da mesma forma, a intensidade de um dado exercício depende de indivíduo para indivíduo, permanecendo esta ao critério do PT.

5.5.2 Zonas de batimentos cardíacos

Como foi referido no subponto anterior, cada intensidade estará diretamente relacionada com um intervalo de batimentos cardíacos ou zona de batimentos cardíacos.

Para alcançar esta zona, é necessário perceber o conceito de batimento cardíaco máximo ou como este irá ser referido, **HRmax**. Isto é, o **HRmax** define o limite máximo de frequência cardíaca de um indivíduo.

Este valor é normalmente obtido através de uma fórmula que tem como a variável a idade do indivíduo, sendo ela:

$$HRmax = 220 - idade$$

Com os avanços realizados em vários estudos sobre este assunto, foi definida uma nova equação para obter o valor mais próximo de um limite máximo. De acordo com o estudo de (Tanaka, Monahan, & Seals, 2000), 237 homens e 277 mulheres com idades entre os 18 e os 81, foram estudados de modo a verificar a integridade da antiga equação. Como resultado, atualizou-se essa fórmula para a seguinte:

$$HRmax = 208 - 0.7 \times idade$$

A fórmula anterior foi utilizada neste projeto para determinar o batimento cardíaco máximo que irá ser necessário para calcular as zonas de batimento cardíacos.

Como foram definidos três tipos de intensidades (baixa, média e alta), será também necessário definir um intervalo de batimentos cardíacos para cada intensidade (Tabela 6 (Ferreira, Coelho, & Nogueira, 2018)).

Tabela 6 - Tabela de zonas de batimentos cardíacos

Zona	Intensidade	Intervalo de HRmax
1	Baixa	65% - 75%
2	Média	76% - 85%
3	Alta	86% - 95%

De seguida, expõe-se um exemplo representativo deste intervalo.

- Sabendo que:
 - **HRmax** = 190;
 - Intensidade desejada é a **média**;

$$\text{Intervalo de HRmax} = [(0.76 \times 190) - (0.85 \times 190)]$$

$$\text{Intervalo de HRmax} = [144 - 161]$$

Portanto, sabendo qual a intensidade do exercício e o *HRmax* do indivíduo, facilmente se sabe o intervalo de batimento cardíaco pretendido. Caso seja necessário mudar de intensidade, haverá a possibilidade de agrupar e filtrar todos os exercícios por uma dada intensidade. Há que referir ainda que, desta forma, facilmente se criam novas intensidades ou se alteram os intervalos de uma já existente.

5.5.3 *Objetivos e Zonas de batimentos cardíacos*

Para a parametrização de um treino, seria necessário definir vários tipos de treino dependendo do objetivo do cliente do PT. De forma a generalizar os tipos de treinos foram estabelecidos **três** tipos de treinos ou objetivos:

- **Loosing fat** (perder gordura);
- **Gaining muscle** (ganhar musculo);
- **Muscular reinforcement** (reforço muscular) (Ferreira, Coelho, & Nogueira, 2018).

Tabela 7 - Representação dos Objetivos em relação com as zonas cardíacas

Id	Objetivo	Zona ideal de batimento cardíaco
1	Perder gordura	2 - 3
2	Ganhar musculo	3
3	Reforço muscular	1 - 2

Com a Tabela 7 é criada uma ligação entre o **objetivo do treino** e as **zonas ideias de batimento cardíaco** que, por sua vez, têm uma relação com a **intensidade**, estando esta última associada a um ou mais **exercícios**.

Assim, estão criadas as entidades necessárias para a elaboração de um plano de treino dinâmico e modular.

5.5.4 HIIT

Existe ainda a possibilidade de estabelecer exercícios em **HIIT** ou **High Intensity Interval Training**, ou seja, criar um circuito de exercícios efetuados a uma alta intensidade durante alguns momentos com tempos de recuperação entre os mesmo (Wewege, Berg, Ward, & Keech, 2017).

Este tipo de treino tornou-se popular, uma vez que é bastante eficiente em termos de tempo, sendo esta uma das principais razões que impede que as pessoas pratiquem exercício físico (Wewege, Berg, Ward, & Keech, 2017).

Foi também estudado que, praticar exercício através do HIIT, aumenta a perda de massa gorda. Apesar disso, existe o inconveniente de o batimento cardíaco estar normalmente muito alto devido à alta intensidade de treino. Neste caso, o PT terá de estar atento ao batimento cardíaco do cliente para este não ultrapassar o batimento cardíaco máximo. Caso este ultrapasse o limite, a intensidade de treino deverá diminuir (Zhang, et al., 2017).

5.5.5 Algoritmo de Geração de Treino

Com base nos capítulos anteriores, será explicado o algoritmo utilizado na implementação da aplicação em questão. Este algoritmo depende das diferentes variáveis referidas, como é o caso da intensidade, dos exercícios, do tipo de treino e, principalmente, do batimento cardíaco que o Apple Watch fornece.

Perante cada batimento cardíaco obtido, o algoritmo avalia o mesmo e valida se este se encontra dentro do **intervalo cardíaco** (subcapítulo 5.5.2) estabelecido para o tipo de treino em questão. Se o batimento cardíaco obtido for **superior** ao intervalo do treino estabelecido, o algoritmo ajusta rapidamente o exercício em questão, **diminuindo a intensidade** do mesmo e vice-versa.

Relativamente ao **batimento cardíaco máximo** (subcapítulo 5.5.2), a cada batimento recebido, é validado se o máximo é atingido. Esta informação é de extrema importância para o PT, uma vez que este poderá estar a lidar com clientes com Insuficiência Cardíaca ou com algum outro tipo de incapacidade cardíaca.

No entanto, o utilizador poderá também obter batimentos cardíacos abaixo do intervalo rítmico da menor intensidade do treino. Neste caso, o PT deve aumentar a intensidade de treino, pois o algoritmo não consegue gerar mais exercícios para o ritmo cardíaco do utilizador.

Por último, o algoritmo agrega o **HIIT** (subcapítulo 5.5.4). Esta variável permite ao algoritmo perceber se o treino é ou não de alta intensidade. Isto é, se o treino for de alta intensidade, o algoritmo não decresce a mesma, avisando apenas o PT quando o utilizador ultrapassou o limite cardíaco máximo, uma vez que estes tipos de treinos são conhecidos por serem bastante exigentes em termos de performance cardíaca.

Quanto à **seleção do exercício**, esta é escolhida na aplicação pelo PT ou é baseada no algoritmo onde, sempre que um exercício termina ou a intensidade do mesmo é alterada (despoletada pelo batimento cardíaco), é escolhido um exercício aleatório da mesma intensidade.

É importante referir que todo este algoritmo é executado enquanto o **tempo do treino não terminar** ou enquanto o PT achar necessário, uma vez que este poderá parar/iniciar o algoritmo quando pretender.

No final, todos os batimentos cardíacos, exercícios executados e informações pertinentes relativas ao treino são guardadas numa lista onde facilmente é possível integrar um módulo de gráficos estatísticos, assim como enviar a informação para o PT via email ou persistir a mesma numa base de dados. Posteriormente, será feita uma análise mais técnica ao algoritmo e à forma como esta lógica foi implementada, sendo necessário haver comunicação entre o iPhone e o Apple Watch.

Toda esta lógica é referida com maior granularidade no documento a que esta dissertação faz referência (Ferreira, Coelho, & Nogueira, 2018).

5.6 Soluções/Abordagens Existentes

Foram analisadas várias abordagens, tendo sido proposta a utilização e desenvolvimento de um dispositivo portátil que se colocaria no interior de uma camisola. Este dispositivo, com recurso aos seus sensores, calcularia os batimentos cardíacos, a temperatura corporal e até a pressão atmosférica, de forma a obter o maior número de informação sobre o utilizador. Para obter a informação deste dispositivo, a aplicação móvel teria de realizar chamadas à respetiva API pública.

A ideia foi descartada pelo facto de criar uma dependência dessa API. Como se tratava de um dispositivo ainda numa fase muito embrionária, isto poderia trazer complicações a longo prazo, uma vez que não seria possível investigar o processo que é utilizado por detrás dessa API.

Uma outra abordagem pensada foi o uso de um outro *wearable*, que poderia trazer uma maior quantidade de informação e assim, acrescentar valor. Um deles seria o Xiaomi Amazfit, mas várias barreiras foram surgindo, como a incapacidade de colocar uma aplicação de terceiros no *Smartwatch* ou a incompatibilidade com a aplicação nativa de iOS Saúde (Amazfit, 2016).

Outra solução proposta seria a criação de uma rede social de PT's e seus clientes, onde estes poderiam ver as pontuações obtidas, criando uma espécie de *marketing* ao PT através das redes sociais. Isto levaria a uma maior concorrência, onde existiria facilidade de acesso aos registos por parte dos clientes, de forma a fomentar a competição e aumentar a sua motivação.

No final, a abordagem seguida foi a descrita nesta dissertação, pelo facto da ITSector não querer ter custos adicionais com novos equipamentos e recursos e também pelo facto de já existirem Apple Watch disponíveis. Tendo em conta que este é o *smartwatch* com maior fiabilidade nos

resultados obtidos, como, por exemplo, o batimento cardíaco (Shcherbina, et al., 2017), foi o escolhido.

Com base neste ponto, foi decidido utilizar o batimento cardíaco para definir planos de treinos dinâmicos, consoante o seu tipo e a intensidade desejada. Estes treinos usariam o batimento cardíaco do indivíduo para averiguar se este se encontra dentro dos limites estabelecidos da intensidade escolhida, ajustando o treino automaticamente conforme o exercício necessitasse de uma maior ou menor frequência cardíaca.

No final, o tempo do PT seria melhor aproveitado, uma vez que este não teria de estudar e desenvolver um plano único para cada um dos seus clientes e tendo em conta que teria acesso a novas informações através do Apple Watch.

Apesar de se ter escolhido esta abordagem, seria **interessante abordar o conceito de rede social**, o que exigiria mais recursos tecnológicos e financeiros por parte da ITSector, uma vez que seriam necessários mais desenvolvimentos na base de dados e no *Middleware*.

5.7 Avaliação das Soluções/Abordagens existentes

Será necessário definir quais os parâmetros determinantes para a avaliação das soluções propostas. Como foi referido na Secção 5.6, os dispositivos primários deste projeto seriam o Apple Watch e o iPhone, por isso, a avaliação em termos técnicos será mais direcionada a estes.

Inicialmente, seria feita uma avaliação à **Usabilidade** das aplicações móveis (iPhone e Apple Watch) recorrendo a um questionário entregue aos utilizadores durante e após o desenvolvimento da aplicação. Desta forma, seria possível garantir que se estaria a seguir o caminho certo e a identificar possíveis melhorias no programa. Segundo Padilha (Padilha, 2004), a técnica do questionário ajuda a encontrar falhas nos projetos e reduz os custos na avaliação da usabilidade, pois é a única forma de conhecer a opinião do utilizador em relação ao programa.

Tabela 8 - Exemplo de Questionário sobre a Usabilidade

	Muito difícil	Difícil	Fácil	Muito fácil
Criar um treino automático, é uma atividade:	x			
Visualizar um gráfico de histórico, é uma atividade:		x		
Manipular o calendário, é uma atividade:			x	
Ler os resultados no Apple Watch, é uma atividade:			x	
A disposição da informação, é de leitura:		x		
...				

Na Tabela 8 está apresentado um exemplo de um questionário possível para avaliar a usabilidade da aplicação, de forma a entender onde se encontram os fluxos mais e menos bem conseguidos.

Com a usabilidade conseguida, seria importante verificar a **Credibilidade e Fiabilidade** dos treinos gerados e da informação obtida, uma vez que o batimento cardíaco é fundamental para a criação automática do plano de treino. A única **forma de verificar a veracidade do objetivo pretendido seria com um grupo de testes realizado num período de tempo de cinco meses a um ano com um PT a utilizar a solução proposta e outro não**. Posteriormente, seria feito um questionário como forma de avaliação do *feedback* dos PT's. Dada a complexidade deste processo, será apenas feita uma referência a uma possível abordagem para a avaliação.

As principais hipóteses a serem abordadas são:

- ✓ **Os planos gerados estão a ser cumpridos?** – é de extrema importância verificar se os planos gerados estarão, de facto, a ser cumpridos. Caso contrário, podem estar a ser de elevada ou de fraca intensidade, sendo que o pretendido é esta nunca estar nos extremos;
- ✓ **A adaptação do treino em tempo real é suficiente?** – o treino será adaptado em tempo real, de acordo com a intensidade e o batimento cardíaco de cada indivíduo. No entanto, esta adaptação pode não ser suficiente, podendo existir a necessidade de criação de mais níveis de treino para estar de acordo com a média dos batimentos cardíacos de certos utilizadores;
- ✓ **O PT obteve melhores resultados usando a aplicação móvel?** – é provavelmente a hipótese mais importante, uma vez que questiona se o PT conseguiu, de facto, obter melhores resultados em termos de agilidade e manutenção do seu dia-a-dia quando recorreu à solução proposta. Esta hipótese está muito dependente das duas acima referidas. Seria necessário realizar um teste de grupo complexo com um período bastante prolongado (de cinco meses a um ano), onde um PT utilizaria a solução proposta e outro não. Posteriormente, seria feito um inquérito de avaliação do *feedback* dos PT's em questão. Por último, com os dados obtidos (relacionados com o tempo gasto e com os treinos gerados com e sem a aplicação), seria realizado um **T-Test** que permitiria comparar as médias de duas amostras e referir qual delas seria a mais vantajosa (T-Test, 2018). No entanto, dada a complexidade deste processo de avaliação, será apenas feita uma referência a uma possível abordagem para o mesmo.

Outra avaliação pertinente para averiguar a **Satisfação** do utilizador seria um processo *rating*, ou seja, à medida que a aplicação fosse utilizada, **surgiria uma pequena janela com cinco estrelas** para o utilizador seleccionar as estrelas correspondentes à sua satisfação, obtendo-se assim um *feedback* geral da solução. Por último, seria realizada uma média dos valores obtidos.

Tabela 9 - Exemplo de *Rating* da Aplicação móvel

	1	2	3	4	5
<i>Em quanto é que avalia o PTrainer?</i>				X	

Em suma, para se obter uma avaliação precisa e fiável, seria necessário realizar um teste complexo que envolvesse, pelo menos, dois PT's e que, no mínimo, um deles utilizasse a aplicação em questão durante algum tempo. Assim, seria feito um questionário de satisfação onde seria detalhado se a solução proposta alcançou ou não os seus objetivos.

5.8 Programas necessários

Para o desenvolvimento do projeto foi necessário utilizar um **Ambiente de Desenvolvimento Integrado** ou **IDE**. Por isso, nesta secção, será feita uma elucidação do programa utilizado, uma vez que este foi a ferramenta mais importante para o desenvolvimento e teste da aplicação.

5.8.1 Xcode

Para o desenvolvimento da aplicação houve a necessidade de utilizar um IDE, ou seja, um programa com o objetivo de maximizar a produtividade dos programadores de software, dando-lhes todas as ferramentas necessárias para garantir softwares de qualidade.

Para desenvolver para os dispositivos da Apple de uma forma nativa, a única solução seria utilizar o IDE oficial chamado **Xcode**. O Xcode permite desenvolver aplicações nativas para Mac OS, iOS, Watch OS e agora TV OS.

Este IDE disponibiliza ferramentas de desenvolvimento que garantem compatibilidade entre todas as versões e modelos do dispositivo para o qual se está a desenvolver, tendo ferramentas de controlo de versões para facilmente submeter uma aplicação para o público (Apple, Xcode 11, 2019).

Tem ainda vantagem de ser completamente grátis, uma vez que qualquer pessoa que tenha acesso a um computador com Mac OS pode utilizá-lo e desenvolver as suas próprias aplicações.

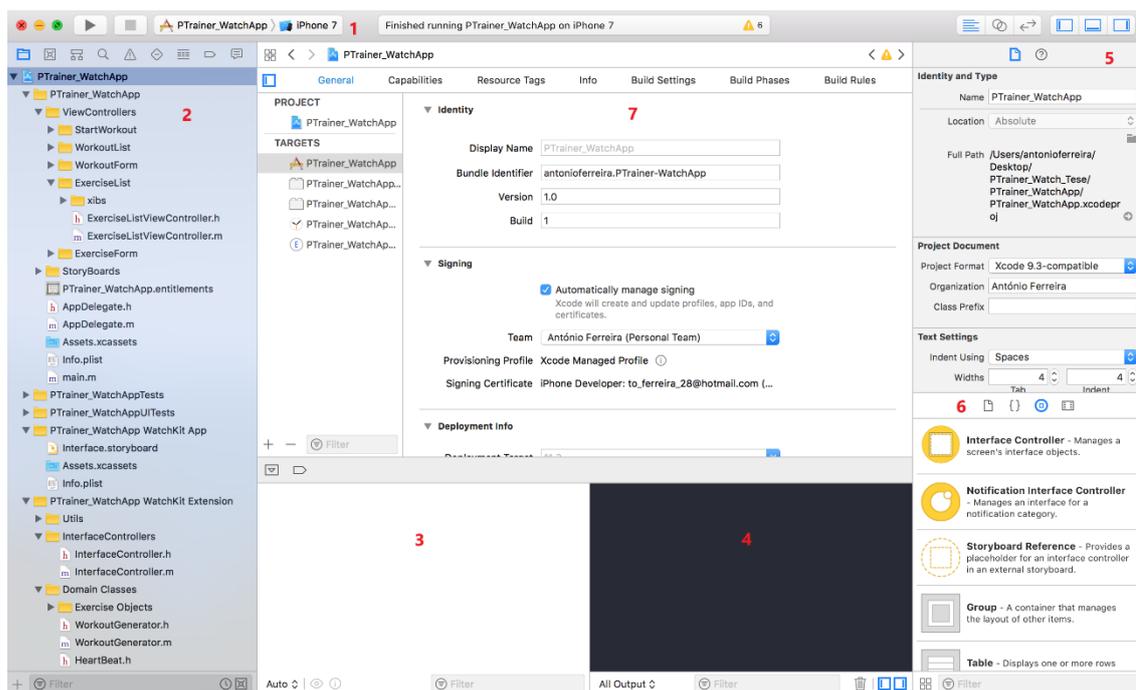


Figura 19 - Estrutura do Xcode

O Xcode é um IDE bastante simples e fácil de manipular, sem necessitar de muito esforço. Encontra-se subdivido em 5 secções principais, onde cada uma tem a responsabilidade de dar informações úteis ao utilizador (Ching, 2018). Essas secções encontram-se expostas de seguida:

1. **Toolbar** – é a secção principal do Xcode, uma vez que **permite construir, executar e terminar a aplicação** num determinado dispositivo. É esta que nos informa do estado da aplicação e dá a possibilidade de manipular a visibilidade do resto das secções, podendo esconder os pontos 2,3,4,5 e 6;
2. **Navigator area** – esta secção fornece a estrutura do/s projeto/s em questão, as pastas, os ficheiros e as configurações. Disponibiliza ainda as instâncias do projeto quando este está em execução, os filtros de pesquisa por código, as lista de **breakpoints** e, entre outros, as ferramentas de consumo de memória;
3. **Debug area** – esta secção fornece todas as instâncias de output do programa, ou seja, sempre que é feito um **debug**, o Xcode disponibiliza o estado de todas as variáveis desta secção;
4. **Debug area** – ainda na mesma secção, mas numa subsecção diferente, permite verificar todo o resultado da aplicação. Assim, todo o **output** gerado pela mesma aparecerá nesta subsecção;
5. **Utility area** – aqui é permitida a manipulação e a verificação de informações sobre qualquer elemento seleccionado, seja este uma classe, objeto, elemento gráfico, ou

ficheiro. Caso seja um elemento gráfico, é possível alterar a cor, da mesma forma que, caso seja um objeto, é possível verificar a documentação do mesmo;

6. **Utility area** – também dentro da secção anterior, esta subsecção permite arrastar elementos gráficos para a *view*, assim como arrastar seções de código fornecidas pelo Xcode ou guardadas pelo utilizador. Tem ainda a possibilidade de verificar todos as imagens (**assets**) da aplicação;
7. **Editor area** – nesta área é visualizado/manipulado todo o código de uma aplicação, sendo possível editar as *views* criadas através dos chamados **Storyboards** (Apple, Storyboard, 2018).

Uma das maiores facilidades que o Xcode fornece é a possibilidade de executar todas as aplicações desenvolvidas em qualquer dispositivo e versão oficial da Apple através de um simulador virtual, o que elimina a necessidade de obter um iPhone ou um Apple Watch.



Figura 21 - Simulador de iPhone 7

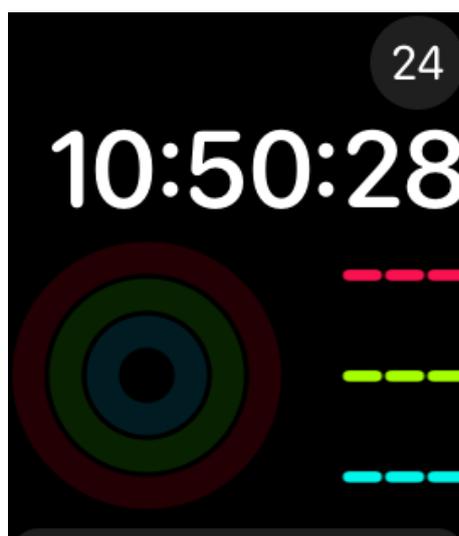


Figura 20 - Simulador de Apple Watch Series 2

Este simulador permite testar as mais comuns funcionalidades como, por exemplo, animações, simulação de GPS, de batimentos cardíacos, de abanão e Touch ID. De qualquer forma, deve sempre ser testado em dispositivos físicos para averiguar os casos de teste cenário real (Ching, 2018).

5.9 Equipamentos necessários

Como foi referido, o Xcode dispõe de simuladores que facilitavam o teste de aplicação em questão tanto para o iPhone como para o Apple Watch. No entanto, para se verificar a fiabilidade do algoritmo de geração de treino, teriam de ser utilizados equipamentos físico. Estes foram fornecidos pela ITSector, facilitando o processo de testes.

5.9.1 iPhone

O iPhone é uma linha de *smartphones* desenhados e publicitados pela Apple. O primeiro iPhone surgiu em 2007, tendo sido lançadas novas versões anualmente após essa data. Isto estabeleceu a marca como uma referência no mercado dos *smartphones*, como se pode observar na Figura 22 (Wikipedia, 2019).

Vários modelos foram desenvolvidos desde o primeiro iPhone (iPhone 3G) até à versão mais recente, o iPhone 11 Pro de 2019. Com versões de maiores dimensões (versão *Max*) ou mais económicas (versão *C*), a Apple tem oferecido várias alternativas ao iPhone dito normal, dando alguma opção de escolha ao consumidor (Apple, iPhone 11 Pro, 2019) (Statista, Unit sales of the Apple iPhone worldwide from 2007 to 2017 (in millions), 2018).

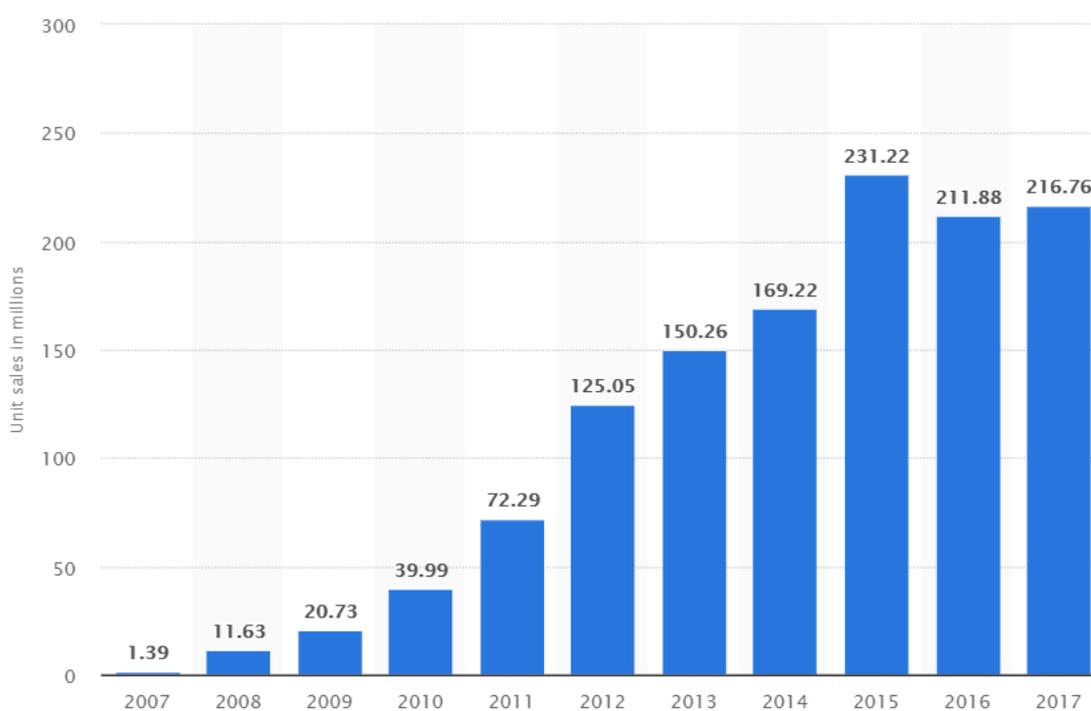


Figura 22 - Vendas de iPhone desde o lançamento do primeiro modelo

O iPhone possui todas as funcionalidades comuns a um *smartphone* como é o caso da câmara, do sistema de mensagens, da loja de aplicações, do GPS, do Bluetooth e do Wi-fi (Apple, iPhone 11 Pro, 2019).

O que diferencia este dispositivo do resto do mercado é o seu software proprietário da Apple e exclusivo para o iPhone e iPad, o **iOS**. Com este sistema operativo, a Apple pretende criar uma experiência fluida, simples e intuitiva num *smartphone*.

Aquando do lançamento de um iPhone, surge sempre uma nova versão do iOS, estando neste momento a caminho da versão 12. Em cada uma das versões são efetuadas melhorias e apresentadas novas funcionalidades.

11.X	91.4%
10.X	5.5%
9.X	1.5%
8.X	0.2%
7.X	1.2%
6.X	0.2%

Figura 23 - Versões do iOS mais utilizadas

Como este sistema operativo é exclusivo para iPhone e é desenvolvido pela Apple, esta aproveita as características dos novos modelos para criar funcionalidades que melhoram a experiência de utilização. Com estas evoluções, surgem novas ferramentas para os programadores manipularem e terem acesso a novas tecnologias como, por exemplo, o **HealthKit** (Apple, HealthKit, 2018) e o **ARKit** (Apple, ARKit, 2018). Isto torna a percentagem de atualizações do sistema operativo extremamente relevantes, como se pode verificar na figura Figura 23 (Smith, 2018), que sugere que cerca de 90% dos utilizadores de iPhone utilizam versões iguais ou superiores ao 11.

No contexto deste documento, para se testar a solução, utilizou-se o iPhone **7** com a versão **11.4** do iOS.

5.9.2 Apple Watch

Com o estatuto criado pela Apple no mercado *mobile* e com a revolução dos *smartwatches* (Zeis, 2014), a Apple introduziu o seu Apple Watch no mercado em 2014. Desde então, este tem vindo a receber várias atualizações a nível de sistema operativo com o WatchOS 6 (Apple, 2019) e a nível físico com o aparecimento do Apple Watch Série 5 em 2019 (Apple, Apple Watch Series 5, 2019).



Figura 24 - Apple Watch modelo simples

- **Como configurar o Apple Watch?**

Para configurar o Apple Watch será, primeiramente, necessário emparelhar o mesmo com um iPhone. Para tal, deve-se aceder à aplicação nativa presente em qualquer iPhone chamada **Watch** (Apple, Set up your Apple Watch, 2018).



Figura 25 - Aplicação Apple Watch

Esta aplicação permite atualizar, configurar e verificar toda a informação do Apple Watch emparelhado. É a partir desta que a configuração inicial do Apple Watch é iniciada, sendo necessário definir o idioma, a password e pré-instalar aplicações de iPhone (Apple, Utilizar apps no Apple Watch, 2018).

Após estes passos, o Apple Watch estará a funcionar corretamente e pronto a ser utilizado.

- **Como instalar uma aplicação no Apple Watch?**

Neste caso, o iPhone estaria emparelhado com o Apple Watch que, por sua vez, estaria diretamente ligado ao Xcode através de um cabo **Lightning** (Apple, Lightning to USB Cable , 2018). Ao executar a aplicação através do **Xcode**, este instala a aplicação em ambos os dispositivos.

Para se realizar uma instalação de forma “oficial”, esta deverá ser instalada primeiramente no iPhone através da **App Store** (Apple, App Store, 2018) da Apple. De seguida, usando a aplicação Watch, é possível instalar a aplicação no Apple Watch, apenas e **só se a aplicação possuir uma versão para Apple Watch** (Apple, Utilizar apps no Apple Watch, 2018).

Para este projeto, a ITSector forneceu um **Apple Watch Série 0** com a última versão do **WatchOS 4**, tendo sido usado maioritariamente para testar o fornecimento de batimentos cardíacos e as suas discrepâncias. Serviu também para testar casos de erro entre o iPhone e o Apple Watch, de forma a avaliar a fiabilidade da aplicação.

5.10 Tecnologias

Neste subcapítulo irá ser realizada uma introdução às tecnologias utilizadas para se atingir o objetivo proposto, nomeadamente a linguagem de programação e as *frameworks* nativas da Apple utilizadas, assim como o envio de informação entre o iPhone e o Apple Watch.

Todos estes conceitos irão servir de introdução teórica para o que será aplicado no capítulo 6 sobre a Implementação.

5.10.1 Objective-C

Objective-C é uma linguagem de programação orientada a objetos desenvolvida por Brad J. Cox, no início de 1980. Tal como o nome indica, é baseada na linguagem de programação procedimental C, tendo obtido mais destaque com a evolução das linguagens orientadas a objetos (Sadun, 2011) (Apple, About Objective-C, 2014).

O Objective-C nem sempre esteve associado à Apple. Efetivamente, começou como uma evolução feita por Cox na linguagem de programação **Smalltalk**, onde este criou uma camada dos objetos do **Smalltalk** sobre a linguagem de programação C. Esta abordagem facilitou o desenvolvimento, uma vez que os programadores poderiam continuar a desenvolver em C e usar objetos através desta nova camada (Sadun, 2011). Em finais de 1980, o Objective-C é adotado pelo sistema operativo NeXTStep de Steve Jobs, como a principal linguagem de programação do sistema. Este é atualmente conhecido como o **Mac OS** ou Mac Operating System, o sistema operativo oficial dos computadores da Apple (Sadun, 2011).

É de notar que o Objective-C evoluiu bastante desde a sua criação. Apesar disso, no desenvolvimento deste projeto utilizou-se a sua última versão.

Como é uma linguagem que herda do C, agrega metodologias algo arcaicas, como a separação da parte pública (.h) e da parte privada (.m) que, de certa forma, auxilia na modularização do código. No entanto, é algo que várias linguagens de programação como o Java, C# ou até o Swift já descartaram para utilizar o **Public/Private** para definir acessibilidade.

Esta é também uma linguagem com uma sintaxe bastante única. Apesar disso, a criação de objetos e de classes e a chamada de métodos pode ser algo confusa para quem começou a estudar, por exemplo, Java.

```
@property (nonatomic, strong) NSArray* bodyAreaArray;
```

Figura 26 - Exemplo de criação de um objeto de classe

Na Figura 26 é possível perceber o nome do objeto (bodyAreaArray) e o seu tipo (**NSArray**), através da sua nomenclatura (**nonatomic, strong**). Estas caracterizam um objeto, aplicando regras de gestão de memória e até de acessibilidade (Apple, Encapsulating Data, 2014).

- **Nonatomic** – especifica quando um objeto ou propriedade pode ser lido/escrito de uma forma rápida. Por outro lado, se a propriedade for acedida por várias instâncias (**threads**) de um programa pode resultar em erros;
- **Atomic** – ao contrário do nonatomic, esta nomenclatura define quando um objeto é seguro para ser acedido por várias instâncias, no entanto, é mais lento;
- **Strong** – define uma ligação forte com quem o cria, ou seja, este objeto só é destruído quando o seu pai for destruído;
- **Weak** – define o contrário do Strong, isto é, o objeto pode ser destruído em tempo de execução, uma vez que o Objective-C entendeu que não é um objeto prioritário.

Mas existem outras definições que podem ser dadas aos objetos como:

- **Readonly** – tal como nome indica, este objeto ou variável só pode ser lida e nunca alterada;
- **Readwrite** – define um objeto que pode ser lido e alterado. Esta nomenclatura pode ser omitida na declaração do objeto ou variável, uma vez que o Objective-C a atribui a todos os objetos por omissão;
- **Copy** – Define que, quando um objeto é alterado, este cria uma cópia do novo valor, atribuindo-o a si mesmo e retendo a lógica de alterar o objeto inacessível;

- **Retain** – também é definido por omissão pelo Objective-C e é gerido pelo mesmo, na medida em que a referência de memória é alterada dinamicamente em tempo de execução, caso esta seja alterada;
- **Assign** – bastante similar ao weak com a diferença de que quando o objeto é destruído, esta ainda pode ter um apontador para a referência de memória onde este está alocado. É algo falível pois esta referência pode já não existir e o programa pode colapsar.

A criação de objetos é algo normal no desenvolvimento de programas. O Objective-C permite definir o estado de vida de um objeto de uma forma fácil e bastante perceptível. Esta caracterização também está visível na criação de classes.

```
@interface SimpleClass : NSObject
...
@end
```

Figura 27 - Parte pública da criação de uma classe.

O Objective-C define no **.h** a sua interface pública. É aqui que se pode definir os objetos públicos de uma classe, ou seja, os objetos que podem ser acedidos por outras classes. Na Figura 27 (Apple, Defining Classes, 2014) é possível observar a sintaxe de criação de uma classe, onde esta se estende a partir do objeto primitivo **NSObject** (Apple, NSObject, s.d.). Caso fosse necessário, existiria a possibilidade de definir outra classe.

Na parte privada (**.m**), existe a possibilidade de recriar a interface como representado na Figura 28, caso haja a necessidade de criar objetivos ou variáveis privadas para serem só acedidas dentro da classe.

```
#import "SimpleClass.h"

@implementation SimpleClass
...
@end
```

Figura 28 - Parte privada de uma classe

É na implementação que são implementados os métodos definidos ou não no ficheiro **.h** da classe. É de notar que, como é uma linguagem baseada no C, para ter acesso a uma classe é necessário importar a mesma através do **Import** que, normalmente, é feito antes de todas as declarações.

Na Figura 29 (Apple, Defining Classes, 2014) é possível verificar que o ficheiro **.m** tem de importar o ficheiro **.h** para este ter acesso às propriedades ou métodos que sejam públicos. A

lógica aplica-se a qualquer classe. Se esta necessitar de certos métodos de uma classe, está obrigada a importar a parte **pública** dessa classe.

```
+(return_type)classMethod{
    // implementation
}
-(return_type)instanceMethod{
    // implementation
}
-(return_type)method:(Type *)parameterName{
    // implementation
}
```

Figura 29 - Estrutura de a declaração de métodos

Para a declaração de métodos, é utilizada uma sintaxe única onde é especificado o **tipo de retorno** entre parênteses e o **nome do método** à frente. É ainda possível especificar **parâmetros** através dos dois pontos onde, de seguida, é necessário definir o tipo do/s parâmetro/s.

Existe ainda a possibilidade de definir se o método é ou não de classe, utilizando o **mais (+)** ou o **menos (-)** respetivamente.

```
[objeto metodo:argumento];
```

Figura 30 - Estrutura de chamada de um método

Para se utilizar um método, terá de se inserir **os parênteses retos** com o nome do objeto em primeiro lugar, procedido de espaço e o nome do método no meio. Se este tiver argumentos deverão ser utilizados os **dois pontos** para a definição dos mesmos.

A utilização de um parâmetro de um objeto é definida pelo **ponto**, semelhante às outras linguagens de programação, dando esta diferenciação entre chamadas de métodos e ao acesso de propriedades de um objeto.

Apesar de ser uma linguagem de programação com uma sintaxe diferente, é bastante segura.

Existem também imensos conteúdos disponíveis oficiais e não-oficiais, como livrarias desenvolvidas por terceiros. Apesar disso, permanece uma linguagem bastante acoplada ao C, o que a limita em termos de novas funcionalidades.

Para colmatar esse ponto, a Apple desenvolveu o **Swift** na versão 5.0, com o objetivo de modernizar e de tornar mais segura a sua linguagem de programação. Ambas estão disponíveis para desenvolvimento e ambas têm a possibilidade de coabitar no mesmo projeto, o que facilita a migração de aplicações de Objective-C para Swift (Apple, Swift 4, 2018).

É importante referir ainda que o Swift é uma linguagem de programação que tem vindo a crescer e que, segundo o TIOBE Index de Agosto de 2018, está em 11º na lista de linguagens de programação com mais adesão, contra o 14º lugar do Objective-C (TIOBE, 2018).

5.10.2 HealthKit

Como foi referido nos subcapítulos anteriores, em cada nova versão do iOS, a Apple anuncia novas funcionalidades e ferramentas para os programadores estudarem e criarem aplicações que tirem proveito destas.

Com o surgir do Apple Watch em 2014 (Apple, Apple Watch Series 5, 2019), surgiu também uma nova tecnologia com o iOS 8 denominada de **HealthKit** (Apple, HealthKit, 2018). Esta tecnologia ou SDK, permite aos programadores integrar todos os dados relativos à saúde do utilizador, através da aplicação nativa no iPhone denominada de **Health**, ou **Saúde** (Apple, Health, 2019).



Figura 31 - Página principal da aplicação Saúde

A aplicação **guarda todo o tipo de informação sobre a saúde do utilizador**, como batimentos cardíacos, tensão arterial, frequência respiratória, índice de massa muscular, percentagem de gordura, quanto tempo o utilizador dormiu, entre muitas outras categorias.

Esta base de dados é consultada através do HealthKit, podendo ler ou escrever informação da mesma, trazendo oportunidades para quem está a desenvolver aplicações e para a Apple, que pretende criar uma base de dados da informação pessoal de saúde de todos os utilizadores (Apple, HealthKit, 2018).

Para além de aceder a estes dados através da integração do HealthKit, é possível criar Workouts ou Treinos (Apple, Workouts, 2018) com o objetivo de aceder, por exemplo, aos batimentos cardíacos em tempo real e fornecer estes dados à aplicação de Saúde. Esta, por sua vez, fornece

os mesmos à chamada **Activity Ring** (Apple, Workouts and Activity Rings, 2018). É de referir que a Apple **não oferece acesso completo aos sensores do Apple Watch**. No caso do cardiofrequencímetro, é utilizado o HealthKit como **interface**, onde este fornece os dados enviados por todos os sensores.



Figura 32 - Exemplo de uma Activity Ring

Esta funcionalidade indica todo o **progresso realizado pelo utilizador** em termos de atividade física, delineando metas e objetivos para este se manter motivado. (Apple, Close Your Rings, 2018). Esta também integrada, maioritariamente, no Apple Watch, enquanto que o iPhone permite a consulta do progresso alcançado.

5.10.3 Watch Connectivity

Desde o início do projeto foi definido que seria necessário que o iPhone comunica-se com o Apple Watch e vice-versa, de forma a que ambos estejam sempre atualizados sobre a informação dos exercícios e a lógica do algoritmo. Por isso, foi necessário realizar uma pesquisa sobre as formas que poderiam definir uma infraestrutura de envio e resposta de mensagens.

Watch Connectivity é um SDK introduzido com o aparecimento do Apple Watch. Este SDK permite estabelecer uma ligação entre o iPhone e o Apple Watch que fornece informações sobre a acessibilidade a ambos os dispositivos. Este SDK mostra se a aplicação do Apple Watch está instalada, se este SDK é suportado pela versão do iOS utilizada e outras formas de envio de informação (Apple, WatchConnectivity, 2018).

A sua estrutura está ilustrada na Figura 6 onde, tal como HealthKit, este SDK é caracterizado como um intermediário, mas, neste caso, entre os dois dispositivos. Isto informa o programador de todas as configurações e eventos que poderão acontecer através do seu delegate denominado de **WCSessionDelegate** (Apple, WCSessionDelegate, 2018).

Como se trata de uma tecnologia oficial da Apple que permite **enviar informação entre os dispositivos de várias formas**, foi estudada a forma de desenvolver um sistema de envio de

informação livre de erros e, preferencialmente, instantâneo, que seria o necessário para implementar a solução proposta.

Posto isto, o Watch Connectivity oferece vários métodos de comunicação com o dispositivo partilhado, nomeadamente:

Tabela 10 - Metodologias de envio de informação entre os dispositivos

Método	Descrição
updateApplicationContext:error:	<p>Envia pequenas quantidades de informação para o dispositivo;</p> <p>O último uso do método substitui o resultado das chamadas anteriores;</p> <p>Não garante que é enviado instantaneamente e usa poucos recursos;</p> <p>Permite envio em background.</p>
transferUserInfo: transferCurrentComplicationUserInfo:	<p>Envia informação em background;</p> <p>O recetor necessita de estar a ser executado para receber essa informação;</p> <p>A informação é enviada em fila;</p> <p>A informação é enviada a qualquer momento para minimizar o consumo de bateria.</p>
transferFile:metadata:	<p>Transfere ficheiros em background;</p> <p>O recetor terá de mudar a localização do/s ficheiro/s recebido/s ou estes serão removidos;</p> <p>A informação é enviada a qualquer momento para minimizar o consumo de bateria.</p>
sendMessage:replyHandler:errorHandler: sendMessageData:replyHandler:errorHandler:	<p>Envia a informação instantaneamente para o recetor;</p> <p>O recetor terá de estar acessível (Reachable (Apple, reachable, 2018)) para receber o conteúdo;</p>

	<p>Ativa a aplicação iOS quando o método é usado na aplicação do Apple Watch;</p> <p>A informação é enviada imediatamente em fila pela ordem de chamada.</p>
--	--

Com base na Tabela 10 (Apple, Sharing Data, 2018) acima referida, é notório o esforço da Apple em oferecer várias metodologias de envio de informação para o uso dos programadores. No entanto, no que toca à solução em questão, faria sentido utilizar o método **sendMessage**, uma vez que permite **enviar mensagens entre os dispositivos em 1, 2 segundos de intervalo de tempo**. Isto é o pretendido, uma vez que a informação necessita de ser enviada o mais rápido possível para não comprometer a fiabilidade da aplicação.

Contudo, é necessário proteger a aplicação de certas situações como, por exemplo, quando os dispositivos perdem a ligação entre si e as mensagens são perdidas. É também necessário salvaguardar a aplicação ao verificar se esta se encontra instalada no Apple Watch. Todas estas características serão **detalhadas de seguida na Implementação**, onde serão dados exemplos da forma como o delegate do Watch Connectivity informa sobre o estado da ligação entre os dispositivos.

6 Implementação

Neste capítulo será dissecado o desenvolvimento técnico, dando ênfase a todas as tecnologias utilizadas para atingir o objetivo proposto. É de notar que o conteúdo técnico exposto não é final, mas está de acordo com os padrões da programação aprendidos na licenciatura e, consequentemente, no mestrado.

6.1 HealthKit

Para a solução apresentada nesta dissertação foi necessário integrar a tecnologia HealthKit que, como foi dito, fornece os batimentos cardíacos do utilizador.

Para tal, foi necessário em primeiro lugar importar este SDK através da seguinte linha:

```
#import <HealthKit/HealthKit.h>
```

Figura 33 - Como importar o HealthKit

Com o SDK integrado, é possível agora aceder a todos os seus métodos e propriedades. De seguida, foi definida uma classe **WorkoutManager** com a responsabilidade de gerir toda a informação recebida pelo HealthKit. Assim, esta seria um intermediário entre quem pretende utilizar estas informações e o HealthKit, modularizando assim esta estrutura e delegando responsabilidades a quem as pretende.

```
@interface WorkoutManager() <HKWorkoutSessionDelegate>

//Main HealthKit object
@property (nonatomic, strong) HKHealthStore* healthStore;

//define the activity type and location
@property (nonatomic, strong) HKWorkoutSession* session;
@property (nonatomic, strong) HKUnit* heartRateUnit;
@property (nonatomic, strong) HKQuery* currentQuery;

@end
```

Figura 34 - Propriedades privadas do WorkoutManager

Esta classe define a propriedade principal denominada de **healthStore** que permite iniciar e terminar os **Workouts**, assim como autorizar o acesso (escrita e leitura) às informações da aplicação de Saúde (Apple, HKHealthStore, 2018).

De seguida, é criada a propriedade que define a sessão de treino chamada de **session**, esta detém a responsabilidade de definir as propriedades do treino, uma vez que o HealthKit necessita de as conhecer. É também esta que controla o treino, ou seja, pausas, paragens e interatividades com o **Activity Ring** (Apple, HKWorkoutSession, 2018).

Por último, foram definidas as propriedades **heartRateUnit** (Apple, HKUnit, 2018) e **currentQuery** (Apple, HKQuery, 2018) que, respetivamente, definem o formato do batimento cardíaco que deve ser devolvido pelo HealthKit e a consulta que é realizada para serem devolvidos os resultados, ou seja, o que se pretende obter e em que data é concretizada a consulta.

Apresentadas as principais propriedades desta classe, será agora analisada a lógica desenvolvida para **iniciar e terminar o requisito de batimentos cardíacos**, assim como para verificar o estado da autorização de leitura dos batimentos cardíacos. No entanto, para perceber a forma como o HealthKit comunica com quem o implementa, é necessário perceber o conceito de **Delegate**.

Existem variadas formas de comunicar a outras classes e objetos que algo ocorreu. Através de, por exemplo, o **NSNotificationCenter** (Apple, NotificationCenter, 2018) que está sempre à escuta de notificações realizadas pela aplicação. No entanto, a Apple fornece uma metodologia chamada de **Delegate**.

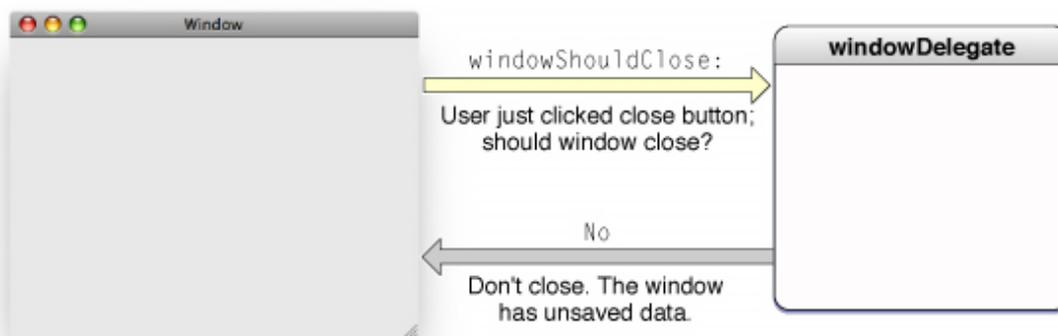


Figura 35 - Como o Delegate funciona

Quando existe uma **classe ou objeto 1** que implementa o **delegate** (Figura 35 (Apple, Delegates and Data Sources, 2012)) de uma **classe 2**, esta é obrigada (ou não, caso seja opcional) a implementar qualquer método que a classe 2 possua. Quando estes são implementados, a classe 2 notifica-os, retornando assim, a informação necessária.

Neste caso, o HealthKit fornece o delegate denominado de **HKWorkoutSessionDelegate** (Apple, HKWorkoutSessionDelegate, 2018) que, quando implementado, notifica sempre que o treino é

iniciado ou sempre que são rececionadas novas informações (Apple, Delegates and Data Sources, 2012).

```
-(void)loadWorkoutManagerSession{
    self.healthStore = [HKHealthStore new];
    self.heartRateUnit = [HKUnit unitFromString:kHeartRateUnit];
    self.currentQuery = [HKQuery new];

    [self checkWorkoutManagerStatus];
}
```

Figura 36 – Método que inicializa propriedades/variáveis

Posto isto, são então inicializadas todas as propriedades anteriormente referidas e é realizada a verificação da autorização de leitura dos batimentos cardíacos através do método **checkWorkoutManagerStatus**.

```
-(void)checkWorkoutManagerStatus{

    if (![HKHealthStore.isHealthDataAvailable]) {
        //verifica se o HealthKit é suportado pelo iPhone em questão
        ...
    }

    if (![HKQuantityType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate]) {
        //verifica se é possível criar o tipo de batimento cardíaco / Heart Rate
        ...
    }

    HKQuantityType *type = [HKQuantityType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];

    [self.healthStore requestAuthorizationToShareTypes:nil
                    readTypes:[NSSet setWithObjects:type, nil]
                    completion:^(BOOL success, NSError * _Nullable error) {
        if (!success) {
            //caso a autorização tenha sido reprovada
        }
    }];
}
```

Figura 37 - Método que verifica o estado do HealthKit e a autorização de acesso aos dados

Neste método são verificadas todas as definições necessárias para o bom funcionamento da requisição de batimentos de cardíacos e do treino, nomeadamente, verificar se o dispositivo suporta o HealthKit (uma vez que este SDK só é suportado a partir do **iOS 8**) e verificar se é possível criar o tipo específico de batimento cardíaco.

Consequentemente, é utilizado o método **requestAuthorizationToShareTypes** do **healthStore** passando os tipos de informação que são pretendidos por parâmetro, sendo que,

automaticamente, surge um ecrã no iPhone a questionar o utilizador se autoriza que a aplicação em questão aceda a estas informações.

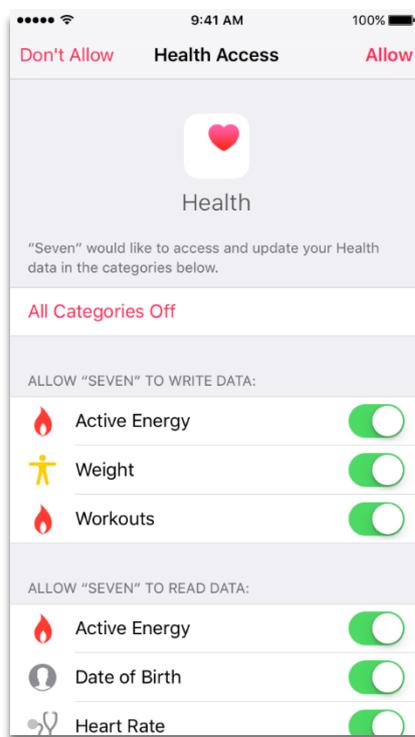


Figura 38 - Exemplo do ecrã de autorização de acesso à informação de saúde

Como se pode observar na Figura 38 (Apple, HealthKit, 2018), o utilizador pode definir se pretende ou não dar acesso a determinadas informações que a aplicação requer. No caso em questão, só é pedido o fornecimento dos batimentos cardíacos.

De seguida, é possível iniciar o treino, criando primeiro uma instância da propriedade **session** para, posteriormente, passá-la como parâmetro para o método **startWorkoutSession** da propriedade **healthStore**, dando início ao treino/workout.

```

-(void)startWorkoutSession{
    // If we have already started the workout, then do nothing.
    if (self.session) {
        return ;
    }

    // Configure the workout session.
    HKWorkoutConfiguration* workoutConfiguration = [HKWorkoutConfigu-
ration new];
    workoutConfiguration.activityType = HKWorkoutActivityTypeOther;
    workoutConfiguration.locationType = HKWorkoutSessionLocation-
TypeIndoor;

    @try {
        self.session = [[HKWorkoutSession alloc] initWithConfigura-
tion:workoutConfiguration error:nil];
        self.session.delegate = self;

        [self.healthStore startWorkoutSession:self.session];
    } @catch (NSEException *e) {
        if([self.delegate respondsToSelector:@selector(workoutManager-
DidFailWithMessage:)]) {
            [self.delegate workoutManagerDidFailWithMessage:e.reason];
        }
    }
}

```

Figura 39 - Método de inicialização do HKWorkoutSession

É importante referir que, aquando da criação da propriedade **session**, é configurada a mesma através da criação do objeto **workoutConfiguration** (Apple, HKWorkoutConfiguration, 2018). A partir desta propriedade, é possível definir o tipo de treino de uma extensa lista de desportos (corrida, natação, entre outros) e a localização do treino, ou seja, se este está a ser executado no interior ou ao ar livre. Estas propriedades são importantes pois classificam os treinos para que estes, quando inseridos na aplicação de Saúde, estejam devidamente organizados.

Para que o treino seja confirmando, o HealthKit terá de informar quem o implementa através do delegate enunciado em cima.

```

-(void)workoutSession:(HKWorkoutSession *)workoutSession
    didChangeToState:(HKWorkoutSessionState)toState
        fromState:(HKWorkoutSessionState)fromState
            date:(NSDate *)date{
    switch (toState) {
        case HKWorkoutSessionStateRunning:
            [self workoutDidStartWithDate:date];
            break;
        case HKWorkoutSessionStateEnded:
            [self workoutDidEndWithDate:date];
            break;
        default:
            NSLog(@"Unexpected state %ld", (long)toState);
            break;
    }
}

-(void)workoutSession:(HKWorkoutSession *)workoutSession didFail-
WithError:(NSError *)error{
    NSLog(@"WORKOUT ERROR");
    if([self.delegate respondsToSelector:@selector(workoutManagerDid-
FailWithError:)]) {
        [self.delegate workoutManagerDidFailWithError:error];
    }
}

```

Figura 40 - Métodos delegate do HKWorkoutSession

Ambos os métodos acima referidos são implementados através do **HKWorkoutSessionDelegate** anteriormente referido. O primeiro método é percorrido sempre que existe alguma alteração do treino, ou seja, **sempre que este começar, for pausado ou terminado**. Já o segundo método é invocado sempre que existe algum erro, geralmente grave, em relação ao treino.

Por isso, o primeiro método é então invocado, dando o feedback que o treino foi inicializado. Perante o sucedido é agora necessário fornecer a consulta do treino, para que este tome conhecimento de quando e como é que é pretendido obter os batimentos cardíacos.

```

-(void)workoutDidStartWithDate:(NSDate *)date{

    if ([self createHeartRateStreamingQueryWithDate:date]) {

        self.currentQuery = [self createHeartRateStreamingQuery-
WithDate:date];
        [self.healthStore executeQuery:self.currentQuery];
    }else{
        if([self.delegate respondsToSelector:@selector(workoutManager-
DidFailWithMessage:)] ) {
            [self.delegate workoutManagerDidFailWithMessage:@"CANNOT
CREATE QUERY"];
        }
    }
}
}

```

Figura 41 - Método que informa o HealthKit da consulta a ser feita para obter dados

O método **workoutDidStartWithDate** verifica se é possível construir uma consulta com a data enviada e se sim, este informa o **healthStore** da consulta a ser executada através do método **executeQuery**.

```

-(HKQuery *)createHeartRateStreamingQueryWithDate:(NSDate*) workoutStartDate {
    HKQuantityType* heartRateType;

    //verificado novamente se é possível criar a propriedade do tipo de batimento cardíaco
    if ([HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate]) {
        heartRateType = [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
    }else{
        return nil;
    }

    //criado o predicado que define a data de início e sem data de fim, ou seja, todos os resultados a partir da data de início
    NSPredicate* datePredicate = [HKQuery predicateForSamplesWithStartDate:workoutStartDate
                                endDate:nil
                                options:HKQueryOptionStrictEndDate];

    NSPredicate* predicate = [NSCompoundPredicate andPredicateWithSubpredicates: [NSArray arrayWithOb-
ject:datePredicate]];

    //é criado uma consulta ancorada utilizando o predicado e o tipo de dados que se pretende obter
    HKAnchoredObjectQuery *heartRateQuery = [[HKAnchoredObjectQuery alloc] initWithType:heartRateType
                                             predicate:predicate
                                             anchor:nil
                                             limit:HKObjectQueryNoLimit
                                             resultsHandler:^(HKAnchoredObjectQuery * _Nonnull query,
NSArry<__kindof HKSample *> * _Nullable sampleObjects,
NSArry<HKDeletedObject *> * _Nullable deletedObjects,
HKQueryAnchor * _Nullable newAnchor,
NSError * _Nullable error) {
        [self updateHeartRate:sampleObjects];
    }];

    // sempre que esta consulta é atualizada através da consulta ancorada, esta atualiza os batimentos cardíacos
    heartRateQuery.updateHandler = ^(HKAnchoredObjectQuery * _Nonnull query,
NSArry<__kindof HKSample *> * _Nullable addedObjects,
NSArry<HKDeletedObject *> * _Nullable deletedObjects,
HKQueryAnchor * _Nullable newAnchor,
NSError * _Nullable error) {
        [self updateHeartRate:addedObjects];
    };

    return heartRateQuery;
}

```

Figura 42 - Criação de uma consulta dando uma data de início e tipo de dados

Com o método da Figura 42 é possível observar como é criada a consulta para se obter batimentos cardíacos a partir da data enviada. Esta data é importante, uma vez que dita quando a consulta deve começar (esta data é referida na aplicação Saúde). Com isto, a condição de pesquisa é definida através de um **NSPredicate** (Apple, NSPredicate, 2018) que é criado para filtrar resultados de uma forma mais leve em termos de memória.

A partir daqui é criado um **HKAnchoredObjectQuery** que permite aceder aos resultados do repositório do HealthKit de forma rápida e fragmentada, referindo objetos desejados e permitindo que os resultados sejam atualizados através do **updateHandler** (Apple, HKAnchoredObjectQuery, 2018).

Ou seja, este objeto é inicializado com o predicado criado anteriormente e com o tipo de dados que pretende procurar. Conforme este recebe resultados, é invocado o método **updateHeartRate** que informa quem implementa o delegate da classe **WorkoutManager** de um novo resultado na consulta.

Finalmente, quando o treino é terminado pelo utilizador, o método na Figura 40 é novamente executado com o novo estado de **HKWorkoutSessionStateEnded**, onde,, posteriormente é terminada toda a lógica de consulta de dados como se pode verificar na Figura 43.

```
-(void)workoutDidEndWithDate:(NSDate *)date{
    if (self.session) {
        [self.healthStore stopQuery:self.currentQuery];
        self.session = nil;
    }
}
```

Figura 43 - Conclusão do treino

É importante referir que toda esta lógica está preparada para qualquer alteração ou adição de funcionalidades. Foi de grande preocupação tornar este módulo de fácil compreensão e de simples escalabilidade.

6.2 Watch Connectivity

Para implementar este SDK na solução, foi optado pela mesma abordagem realizada com o HealthKit, ou seja, criar uma classe intermediária encarregue de filtrar a informação vinda do delegate nativo e as várias configurações, sem a necessidade de implementar sempre que se pretender aceder a estes métodos.

Como tal, será primeiro necessário importar este SDK para ter acesso a todos os seus métodos e ao seu delegate.

```
#import <WatchConnectivity/WatchConnectivity.h>
```

Figura 44 - Importação do Watch Connectivity

Com a importação realizada, é possível agora ter acesso a todos os métodos públicos, mas, antes disso, foi primeiro necessário decidir todas as propriedades privadas necessárias para a lógica na classe **CommunicationManager**. No entanto, só seria necessário utilizar uma propriedade chamada **messagingSession** do tipo **WCSession** (Apple, WCSession, 2018), pois é esta que define todas as configurações da ligação entre o Apple Watch e o iPhone e todos os métodos necessários para ativar e desativar a mesma.

```
@interface CommunicationManager() <WCSessionDelegate>

//Variables responsible for creating the communication between
iphone and apple watch
@property (nonatomic, strong) WCSession* messagingSession;

@end
```

Figura 45 - Propriedades privadas da classe Communication Manager

Posto isto, é agora necessário inicializar a propriedade **messagingSession** e definir um método para verificar o estado da comunicação entre os dispositivos.

O método de inicialização é atualmente público para existir a possibilidade de lhe ser enviado como parâmetro o implementador, ou seja, quem o pretende implementar, de forma a garantir a modularidade da implementação.

No entanto, é o **CommunicationManager** que implementa os métodos do Watch Connectivity e filtra a informação que, por sua vez, é enviada para quem implementa o delegate **CommunicationManagerDelegate** criado nesta classe para comunicar a existência de mudanças e de mensagens recebidas.

```

-(void)loadCommunicationManagerWithDelegate:(id)delegate{
    if ([WCSession isSupported]) {
        self.messagingSession = [WCSession defaultSession];
        if (delegate) {
            self.messagingSession.delegate = delegate;
        }else{
            self.messagingSession.delegate = self;
        }
        [self.messagingSession activateSession];
    }
}

-(void)checkSessionStatus{
    if (![self.messagingSession isReachable]) {
        //iPhone and Apple Watch are NOT reachable
    }else{
        //iPhone and Apple Watch ARE reachable
    }
}

```

Figura 46 - Métodos de inicialização e verificação do estado da ligação entre os dispositivos

Como é possível analisar na Figura 46, é primeiro verificado se esta ligação é suportada através do método público **isSupported** (Apple, isSupported, 2018). De seguida, através do método **defaultSession** (Apple, DefaultSession, 2018), é inicializada a propriedade responsável pela sessão. Este método retorna sempre à mesma sessão, uma vez que se trata de um **singleton** (W3sDesign, 2018).

Posteriormente, é verificado se existe algum implementador do delegate. Caso seja, a classe implementa diretamente os métodos do WatchConnectivity, no entanto, caso não seja enviado, será de a responsabilidade do CommunicationManager filtrar a informação e enviá-la para quem o implementa. Por último, é ativada a sessão entre os dispositivos através do método **activateSession** (Apple, ActivateSession, 2018).

Quanto ao segundo método, este verifica se os dispositivos possuem uma ligação entre eles. Este método é público e poderá ser executado sempre que a classe que implementa deseja verificar a ligação dos dispositivos.

```

-(void)sendWakeMessage{
    [self.messagingSession sendMessage:
        [[NSDictionary alloc] initWithObjects:@[@"Wake up!"]
        forKeys:@[kWakeUp]]
        replyHandler:nil
        errorHandler:nil];
}

-(void)sendMessageWithKeys:(NSArray<NSString*>*)keys
    andValues:(NSArray<NSObject*>*)objects{
    if ([self.messagingSession isReachable]) {
        [self.messagingSession sendMessage:[NSDictionary alloc] initWithObjects:objects
        forKeys:keys]
        replyHandler:^(NSDictionary *reply) {
            [self communicationManagerDidReplyWithMessage:reply];
            //handle reply from iPhone app here
        }
        errorHandler:^(NSError *error) {
            //catch any errors here
        }
    }
}
}
}

```

Figura 47 - Métodos de envio de mensagens

Para a comunicação foi criado o método **sendMessageWithKeys** que envia mensagens através do método do WatchConnectivity chamado **sendMessage** (Apple, sendMessage, 2018). Em cada mensagem existe uma chave que é conhecida pelos dois dispositivos, sendo possível através desta identificar as mensagens de forma a executar funcionalidades específicas (Figura 48). Este método permite ainda manipular a/resposta/as que pode/m ou não ser enviada/s, **tal como verificar algum erro que possa surgir com o envio**, notificando a classe que a implementa do sucedido.

É de notar que para a informação ser enviada, esta tem de ser transformada em formato de **texto ou em JSON**, de forma que o outro dispositivo consiga reconstruir a mesma para um objeto e, assim, ser utilizado de forma habitual, uma vez que não é possível enviar objetos ou propriedades que não sejam primitivas.

```

static NSString * const kStartWorkout = @"StartWorkout";
static NSString * const kResumeWorkout = @"ResumeWorkout";
static NSString * const kPauseWorkout = @"PauseWorkout";
static NSString * const kStopWorkout = @"StopWorkout";
static NSString * const kWorkoutStarted = @"WorkoutStarted";
static NSString * const kRestTime = @"RestTime";
static NSString * const kSkipExercise = @"SkipExercise";
static NSString * const kPreviousExercise = @"PreviousExercise";
static NSString * const kHeartRate = @"HeartRate";
static NSString * const kHeartRateList = @"HeartRateList";
static NSString * const kExerciseChanged = @"ExerciseChanged";
static NSString * const kDecreaseIntensity = @"DecreaseIntensity";
static NSString * const kIncreaseIntensity = @"IncreaseIntensity";
static NSString * const kIntensity = @"Intensity";
static NSString * const kMaximumHeartRate = @"MaximumHeartRate";
static NSString * const kMaximumHeartRateReached = @"MaximumHeartRateReached";
static NSString * const kMinMaxHeartRateInterval = @"MinMaxHeartRateInterval";
static NSString * const kWorkoutTime = @"WorkoutTime";
static NSString * const kHiit = @"HIIT";
static NSString * const kWakeUp = @"WakeUp";

```

Figura 48 - Chaves de identificação de mensagens

De forma a melhorar a usabilidade da aplicação, foi criado um método onde, sempre que a aplicação do Apple Watch é iniciada, esta envia uma mensagem ao respetivo iPhone, ordenando-o a iniciar a aplicação automaticamente em background.

Posto isto, serão agora referidos os delegates da própria classe e o Watch Connectivity, pois são estes que informam o estado da ligação e que informação foi rececionada.

```
-(void)communicationManagerDidReplyWithMessage:(NSDictionary*)message{
    if([self.delegate respondsToSelector:@selector(communicationManagerDidReplyWithMessage:)]) {
        [self.delegate communicationManagerDidReplyWithMessage:message];
    }
}

-(void)communicationManagerDidReplyError:(NSError*)error{
    if([self.delegate respondsToSelector:@selector(communicationManagerDidReplyError:)]) {
        [self.delegate communicationManagerDidReplyError:error];
    }
}

-(void)communicationManagerDidChangeToState:(NSString*)message{
    if([self.delegate respondsToSelector:@selector(communicationManagerDidChangeToState:)]) {
        [self.delegate communicationManagerDidChangeToState:message];
    }
}
```

Figura 49 - Métodos do CommunicationManagerDelegate

Começando pelo delegate da própria classe o **CommunicationManagerDelegate**, este possui métodos que, se implementados pela classe que implemente o WorkoutManager, informam esta **se ocorreu alguma resposta a uma mensagem enviada**, se existiu algum erro no envio da mesma ou se o estado da ligação foi modificado por algum motivo.

Para além destes, existem também os métodos fornecidos pelo WatchConnectivity (Figura 50 (Apple, WCSSessionDelegate, 2018)) que detalham com maior rigor o estado da ligação (se o iPhone se emparelhou com vários dispositivos).

```

- (void)session:(nonnull WCSSession *)session activationDidCompleteWithState:(WCSession-
ActivationState)activationState error:(nullable NSError *)error {
    if([self.delegate respondsToSelector:@selector(communicationManagerDidChangeS-
tate:)]) {
        [self.delegate communicationManagerDidChangeState:activationState];
    }
}

-(void)session:(WCSession *)session didReceiveMessage:(NSDictionary<NSString *,id>
*)message replyHandler:(void (^)(NSDictionary<NSString *,id> * _Nonnull))replyHandler{
    replyHandler(@{ @"Status" : @"OK" });

    if([self.delegate respondsToSelector:@selector(communicationManagerDidReceiveMes-
sage:)]) {
        [self.delegate communicationManagerDidReceiveMessage:message];
    }
}

-(void)sessionReachabilityDidChange:(WCSession *)session{
    if([self.delegate respondsToSelector:@selector(communicationManagerReachabilityDid-
Change:)]) {
        [self.delegate communicationManagerReachabilityDidChange:session];
    }
}

//Apenas disponiveis no iOS

= (void)sessionDidBecomeInactive:(nonnull WCSSession *)session {
    if([self.delegate respondsToSelector:@selector(communicationManagerDidBecomeInac-
tive:)]) {
        [self.delegate communicationManagerDidBecomeInactive:session];
    }
}

= (void)sessionDidDeactivate:(nonnull WCSSession *)session {
    if([self.delegate respondsToSelector:@selector(communicationManagerDidDeactivate:)]) {
        [self.delegate communicationManagerDidDeactivate:session];
    }
}

```

Figura 50 - Métodos delegate do WCSSessionDelegate

O método **activationDidCompleteWithState** (Apple, session:activationDidCompleteWithState:error:, 2018), informa **o estado da ligação entre os dois ou mais dispositivos**, sendo útil para a verificação do estado antes de iniciar a lógica do algoritmo proposto por esta dissertação.

Relativamente ao método **didReceiveMessage** (Apple, session(_:didReceiveMessage:replyHandler:), 2018), este receciona as mensagens que são identificadas pelas chaves estabelecidas anteriormente (Figura 48). Para além disso, permite, se necessário, responder às mesmas. No entanto, neste caso, isto não se verifica, uma vez que as mensagens servem apenas para enviar informação.

Já o método **sessionReachabilityDidChange** (Apple, sessionReachabilityDidChange, 2018), é executado pelo WatchConnectivity sempre que a aplicação do Apple Watch entra em Background. Este aspeto será detalhado com maior granularidade no subcapítulo 6.2.1, uma vez que originou um impedimento no desenvolvimento da solução. Assim, **sempre que o ecrã permanece bloqueado e que os dispositivos perdem ligação entre si**, é de extrema importância estar à escuta deste método pois pode existir algum problema com a ligação e a sessão terá de terminar para evitar comportamentos anómalos.

Por último, o WatchConnectivity obriga à implementação dos dois últimos métodos: o **sessionDidBecomeInactive** (Apple, sessionDidBecomeInactive, 2018) e o **sessionDidDeactivate** (Apple, sessionDidDeactivate, 2018). Ambos os métodos têm como propósito informar se a sessão ou a ligação entre os dispositivos emparelhados ficou inativa ou desativada. Isto acontece apenas no caso de existir mais do que um Apple Watch emparelhado com um só iPhone, como se pode verificar na Figura 51.

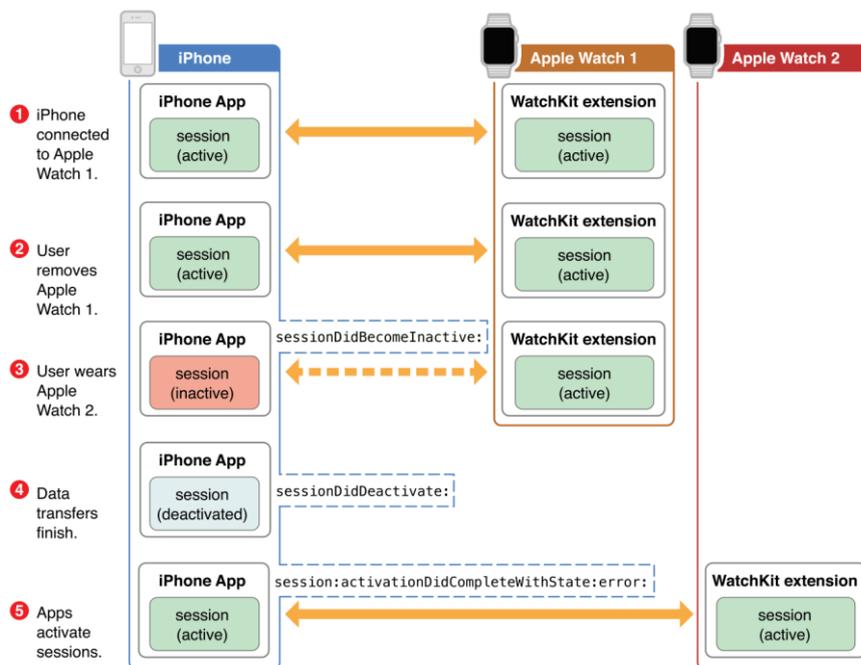


Figura 51 - Estado da sessão quando vários dispositivos estão emparelhados com o mesmo iPhone

Esta lógica é semelhante em iOS e em WatchOS, exceto algumas propriedades que estão definidas apenas em um dos sistemas operativos, tal como o **isWatchAppInstalled** (Apple, isWatchAppInstalled, 2018) que apenas está disponível em iOS. Da mesma forma, existem métodos delegate que podem deter terminologias diferentes nos sistemas operativos, no entanto, ambos estão a aguardar alterações no estado da ligação para que, de forma segura, a ligação seja terminada (Apple, WCSession, 2018).

6.2.1 Ciclo de vida de uma aplicação para Apple Watch

De forma a explicar de uma maneira mais sucinta o que implica a mudança da propriedade **isReachable** do **WatchConnectivity**, é necessário primeiro entender o ciclo de vida de uma aplicação.

O Apple Watch, tal como todos dispositivos que permitem executar aplicações, possuem um algoritmo que gere a memória do mesmo. Nada deve deteriorar a experiência de utilização do dispositivo, e por isso o ciclo de vida de uma aplicação permite que a aplicação quando por

exemplo, está em **background**, ou seja, não visível para o utilizador, esta aloque menos recursos e consumo de memória. Essa memória é realocada para a aplicação que estiver visível (**foreground**). Este processo é ilustrado de uma forma mais visual na Figura 52 (Apple, The Watch App Architecture, 2018).

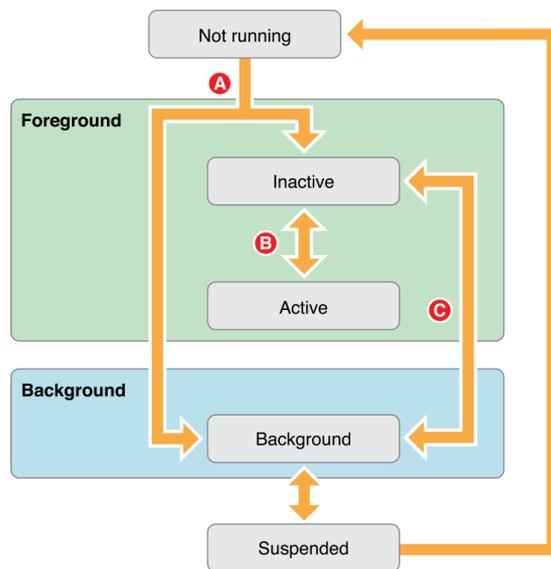


Figura 52 - Ciclo de vida de uma aplicação para Apple Watch

- A. Momento em que a aplicação acabou de ser lançada e executa o método **applicationDidFinishLaunching** (Apple, applicationDidFinishLaunching, 2018);
- B. Momento em que a aplicação está ativa e é visível pelo utilizador (foreground) e executa o método **applicationDidBecomeActive** (Apple, applicationDidBecomeActive, 2018), ou momento em que a aplicação está prestes a ser desativada e executa o método **applicationWillResignActive** (Apple, applicationWillResignActive, 2018);
- C. Momento em que a aplicação não está visível para utilizador (background) e passa a estar visível para o utilizador, executando o método **applicationWillEnterForeground** (Apple, applicationWillEnterForeground, 2018) e o momento em que a aplicação está visível para o utilizador (foreground), mas este entra noutra aplicação, passando-a para background, executando o método **applicationDidEnterBackground** (Apple, applicationDidEnterBackground, 2018).

No caso, sempre que a aplicação entra em background, a propriedade **isReachable** é modificada para **falso**, o que impossibilita o envio de mensagens através do WatchConnectivity, na medida que este obriga a que a propriedade **isReachable** esteja a **verdadeiro** (Apple, sendMessage, 2018).

No contexto do projeto, o cliente do PT que usará a aplicação não estará sempre com a aplicação em foreground, uma vez que basta o utilizador mover o pulso para baixo (Apple Watch virado para o chão) para esta entrar em modo background com o objetivo de consumir menos bateria.

PTTrainer_WatchApp > PTTrainer_WatchApp WatchKit Extension > Info.plist > No Selection

Key	Type	Value
▼ Information Property List	Dictionary	(12 items)
Localization native development region	String	\$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	PTTrainer_WatchApp WatchKit Extension
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▶ NSExtension	Dictionary	(2 items)
▼ Required background modes (Watch)	Array	(1 item)
Item 0	String	Workout Processing
WKExtensionDelegateClassName	String	ExtensionDelegate

Figura 53 - Permitir modo Background no Apple Watch

Para tornar o envio de mensagens em background possível, foi necessário adicionar a permissão para executar a aplicação em background em ambos os dispositivos (Figura 53). O que na prática permite que estes tenham duas sessões da aplicação a serem executadas, uma em background e outra em foreground, o que se torna bastante dispendioso em termos de memória, mas seria a única metodologia para a aplicação enviar informação em tempo real, sem existir quebras em tempo de execução.

6.3 Algoritmo de Geração de Treino (Parte técnica)

Terminada a explicação mais técnica sobre as tecnologias usadas dentro do Objective-C, será agora feita uma elucidação sobre a utilização destas tecnologias para se concretizar o que é proposto por esta dissertação.

Na Figura 54 é possível observar, de uma forma geral, as principais classes da solução com o **CommunicationManager**, responsável pela comunicação entre o iPhone e o Apple Watch, assim como do **HealthKitManager**, responsável pela comunicação do Apple Watch e iPhone com o HealthKit.

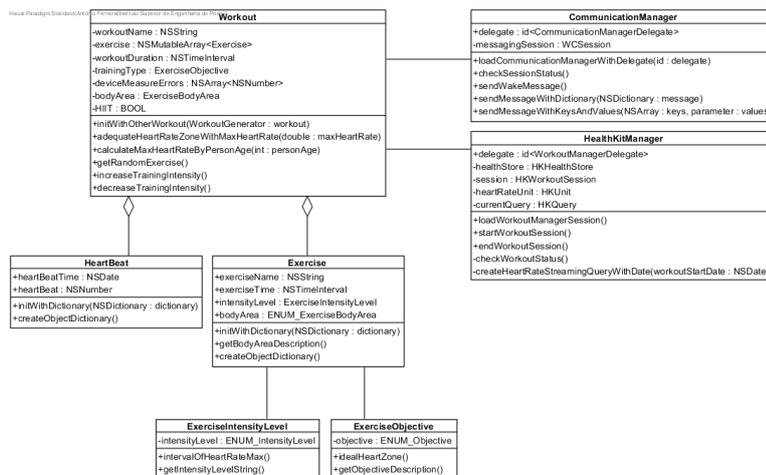


Figura 54 - Diagrama de Classes

Posto isto, é necessário referir que a responsabilidade da utilização deste algoritmo é do Apple Watch, ou seja, é este dispositivo que agrega a lógica do mesmo, dado que contém a informação relativa aos batimentos cardíacos. O iPhone é apenas responsável pela criação e gestão do treino, assim como da requisição de informação do Apple Watch.

Foi optado também pelo envio de pequenas quantidades de informação para o Apple Watch, garantindo que é enviado exercício a exercício, conforme a manipulação do PT. Esta informação indica se o treino é legível para HIIT, o tempo de treino, o batimento cardíaco máximo (calculado através da fórmula já referida no subcapítulo 5.5.2) e o intervalo de batimento cardíaco mínimo e máximo.

```

//Defines first exercise
[self changeExercise];

//sends to watch the necessary data to start the workout
[self.messagingSession sendMessageWithKeys:@[kHiit]
andValues:@[NSNumber
numberWithBool:self.selectedWorkout.HIIT]];

[self.messagingSession sendMessageWithKeys:@[kWorkoutTime]
andValues:@[NSNumber numberWithDouble:
self.selectedWorkout.workoutDuration]]];

[self.messagingSession sendMessageWithKeys:@[kMaximumHeartRate]
andValues:@[NSNumber numberWithInteger:
self.maximumHeartRate]]];

self.minMaxHeartRateInterval = [self.selectedWorkout
adequateHeartRateZoneWithMaximumHeartRate:self.maximumHeartRate];

[self.messagingSession sendMessageWithKeys:@[kMinMaxHeartRateInterval]
andValues:@[self.minMaxHeartRateInterval]];

[self.messagingSession sendMessageWithKeys:@[kStartWorkout]
andValues:@[[self.selectedWorkout.exercises
objectAtIndex:self.selectedExerciseIndex.row] createObjectDictionary]];
  
```

Figura 55 - Informação enviada para Apple Watch ao iniciar treino

O Apple Watch calcula os batimentos cardíacos através da biblioteca nativa da Apple (já referida no subcapítulo 6.1) que, com as características do treino e do exercício a ser executado no momento, avalia-o e responde ao iPhone, referindo o batimento cardíaco e se é ou não necessário mudar de intensidade.

Recebendo a informação do iPhone, o Apple Watch prepara todos os mecanismos lógicos de treino (tempo de treino, tempo de exercício, intensidade de exercício, etc.) e de consulta de batimentos cardíacos, inicializando todas as variáveis necessárias.

No entanto, para isso seria primeiro necessário criar um **temporizador que a cada segundo verificasse estes valores** incluindo variáveis do treino como o HIIT e o batimento cardíaco máximo.

```
self.everySecondTimer = [PausableTimer timerWithTimeInterval:1.0
                        target:self
                        selector:@selector(everySecondAction)
                        userInfo:nil
                        repeats:YES];
```

Figura 56 - Temporizador responsável por chamar o método "everySecondAction"

Este temporizador a cada segundo executa o método **everySecondAction**, que avalia toda a informação do treino e do batimento cardíaco. Este temporizador (PausableTimer) é um módulo criado para permite pausar e resumir o mesmo, uma vez que o NSTimer nativo do iOS não inclui estas funcionalidades e houve a necessidade de criar algo mais complexo para este contexto.

Como o iPhone agrega a informação relativa ao treino, o Apple Watch apenas envia mensagem ao iPhone comunicando se é, ou não necessário alterar a intensidade do treino.

```
if ([self.currentHeartRate.heartBeat doubleValue] >
    [self.minMaxHeartRateInterval.lastObject doubleValue]) {
    if (self.isHIIT) {
        if (self.popUpFlag) {
            ""
        }
    }else{
        [self.messagingSession sendMessageWithKeys:@[kDecreaseIntensity]
                                             andValues:@[@"Decrease the
intensity!"]];
    }
}
else{
    if ([self.currentHeartRate.heartBeat doubleValue] <
        [self.minMaxHeartRateInterval.firstObject doubleValue]) {
        [self.messagingSession sendMessageWithKeys:@[kIncreaseIntensity]
                                             andValues:@[@"Increase the
intensity!"]];
    }
}
```

Figura 57 - Método everySecondAction

Pelo que se pode observar na Figura 57, é realizada a verificação ao **currentHeartRate** que sempre que recebe um novo valor, este é atualizado garantindo que o batimento é atualizado e verificado.

Este batimento cardíaco é primeiro comparado com o batimento cardíaco mais alto do intervalo de batimentos cardíacos (**minMaxHeartRateInterval**) que é estabelecido pela intensidade de treino. Caso afirmativo, é enviada mensagem ao iPhone para decrescer a intensidade de treino. Existe ainda o caso de que se a funcionalidade de HIIT estiver ativa, tanto o utilizador do Apple Watch, como o PT são informados através de uma notificação.

Caso não seja verificada esta condição, é analisado logo de seguida o contrário, ou seja, é comparado o batimento cardíaco com o mínimo do intervalo de batimentos cardíacos, que posteriormente envia mensagem ao iPhone requisitando um aumento de intensidade.

```
//Only heartRates > 0 are valid
    if ([currentHeartRate.heartBeat doubleValue] > 0) {

        [self startTimersMajorLogic];

        //Verifies if heartRate reaches the Maximum
        if ([currentHeartRate.heartBeat doubleValue] > self.maximumHeartRate) {
            ...
            [self.messagingSession sendMessageWithKeys:@[kMaximumHeartRateReached]
                andValues:@[@"🚨🚨🚨 Maximum Heart rate has
been reached! 🚨🚨🚨"]];
        }

        self.popUpFlag = YES;

        [self.heartRateHistory addObject:[currentHeartRate createObjectDictionary]];
        [self.heartRateLabel setText:[NSString
stringWithFormat:@"%f", [currentHeartRate.heartBeat doubleValue]]];

        [self.messagingSession sendMessageWithKeys:@[kHeartRate]
            andValues:@[[currentHeartRate
createObjectDictionary]]];

    }
}
```

Figura 58 - Verificação do Batimento cardíaco

É de igual importância referir que sempre que um batimento cardíaco é obtido, este é **verificado comparando-o com o batimento cardíaco máximo** do dado sujeito, que posteriormente é registado numa lista e enviado imediatamente para o iPhone (Figura 58).

Com esta informação validada pelo Apple Watch e com as mensagens que este envia, é agora necessário referir que lógica o iPhone executa quando estas são recebidas.

```
-(BOOL)increaseTrainingIntensity{
    if (![self.selectedWorkout increaseTrainingIntensity]) {
        return NO;
    }

    self.maximumHeartRate = [self.selectedWorkout
        calculateMaximumHeartRateByPersonAge:kPersonAge];

    //updates de HRMin and HRMax
    self.minMaxHeartRateInterval = [self.selectedWorkout
        adequateHeartRateZoneWithMaximumHeartRate:self.maximumHeartRate];

    return YES;
}
```

Figura 59 - Método de aumento de Intensidade do treino

Primeiramente, o algoritmo tenta aumentar (neste caso) a intensidade do treino, ou seja, aumentando o **Objetivo** de treino (Tabela 7). Caso já esteja no máximo, informa o PT para aumentar a intensidade do exercício atual. No entanto, se houver uma mudança de intensidade com sucesso, é de seguida calculado novamente o **batimento cardíaco máximo**, atualizando a **margem de erro deste novo máximo** (Figura 60).

```
-(int)calculateMaximumHeartRateByPersonAge:(int)personAge{
    int maximumHeartRate = 208 - 0.7 * personAge;

    //Adjusts the MaximumHeartRate by a predefined percentage
    return [self adjustErrorWithMaximumHeartRate:maximumHeartRate];
}
```

Figura 60 - Método de Calculo do batimento cardíaco máximo

Para finalizar a lógica do aumento de intensidade, é calculado os **intervalos mínimos e máximos cardíacos**, para assim o algoritmo permanecer com a lógica de comparação dos batimentos cardíacos obtidos com este intervalo. Para isso é necessário informar o algoritmo de treino do batimento **cardíaco máximo** calculado atrás para este obter os intervalos de percentagem da dada intensidade de treino (subcapítulo 5.5.2).

```

- (NSArray*) adequateHeartRateZoneWithMaximumHeartRate: (double) maximumHeartRate {
    // First Position - Minimum HeartRate
    // Last Position - Maximum HeartRate
    NSArray * array = [NSArray new];
    NSArray<ExerciseIntensityLevel *> * heartZones = [self.trainingType
idealHeartZone];

    if (heartZones.count > 1) {
        array = @[@(maximumHeartRate * [[heartZones.firstObject
intervalOfHeartRateMax].firstObject doubleValue] ),
                @(maximumHeartRate *
[[heartZones.lastObject intervalOfHeartRateMax].lastObject doubleValue] )];
    } else {
        array = @[@(maximumHeartRate * [[heartZones.firstObject
intervalOfHeartRateMax].firstObject doubleValue] ),
                @(maximumHeartRate *
[[heartZones.firstObject intervalOfHeartRateMax].lastObject doubleValue] )];
    }

    NSLog(@"🚩 Minimum and Maximum Heart Rate Intervals 🚩: %@", array);

    return array;
}

```

Figura 61 - Cálculo no novo mínimo e máximo cardíaco

Por último, é **multiplicado o batimento cardíaco máximo** com o **mínimo e máximo** obtidos através do método **idealHeartZone**, que no fim é retornado e informa o algoritmo se existe a necessidade, ou não de mudar de exercício. Caso afirmativo, é enviado o exercício para o Apple Watch, continuando assim o treino. A mesma lógica se aplica quando se decresce a intensidade de treino.

Será **considerado ainda um pequeno ajuste na mudança de intensidade**, uma vez que esta mudança é despoletada logo após ao batimento cardíaco estar fora do intervalo. Este acontecimento pode não ser representativo da realidade, uma vez que o Apple Watch pode obter um batimento cardíaco bastante fora do normal. Para isso, será implementada uma pequena lógica, em que apenas no **segundo/terceiro** batimento cardíaco fora do intervalo é que despoleta a mudança de intensidade, mantendo assim a fiabilidade da aplicação.

7 Conclusões

Esta dissertação teve como objetivo documentar a visão e o propósito do projeto PTrainer, que tem como missão melhorar e automatizar o dia-a-dia de um PT, dando uso às tecnologias existentes que, muitas vezes, são menosprezadas pelo facto de criarem mais barreiras do que facilitismos. Para tal, primeiramente foi necessário estudar o mercado *fitness* e perceber o porquê de as pessoas não utilizarem os serviços do PT, como se pode verificar no subcapítulo 2.1.

Existem várias razões, sendo a principal, sem dúvida, o fator **económico**, uma vez que, para muitas pessoas que apenas pretendem fazer um pouco de exercício, o gasto não se justifica. Existe ainda outro fator não menos importante, as **aplicações gratuitas**. Estas fornecem planos de treinos e demonstrações para os seus utilizadores que, de certa forma, diminuem ainda mais as razões para recorrer aos serviços de um PT.

Desta forma, foi necessário realizar uma profunda pesquisa sobre o estado de arte (capítulo 4) dos *wearables* existentes mais reconhecidos e de algumas das aplicações mais utilizadas, de forma a tentar perceber o que leva os utilizadores a preferirem algumas aplicações em relação a outras e como se poderia utilizar os recursos de um *wearable* para tentar promover o PT na área do *fitness*. Várias aplicações oferecem uma panóplia de exercícios com demonstrações através de vídeo ou de imagem, assim como planos de treino de acordo com o objetivo do utilizador e com ligações aos *wearables*, como é o caso do Samsung Gear Fit, Xiaomi Mi Band, Apple Watch. Onde por fim, foi realizada uma comparação entre os vários dispositivos e as várias aplicações no subcapítulo 4.4, que de uma forma geral serviu para aglomerar as funcionalidades mais interessantes de serem implementadas para melhorar o acompanhamento do PT e os seus clientes.

Houve a preocupação de tentar perceber como se poderia aliar recursos tecnológicos com o PT, e, de certa forma, incentivar o PT a recorrer aos mesmos para aliciar o seu mercado que, muitas vezes, recorre a aplicações gratuitas. Seria necessário criar algo que automatizasse os processos sem criar obstáculos, dando uso a um dos utensílios mais utilizados nos dias de hoje: o *smartphone*.

O *smartphone* veio revolucionar a forma como acedemos à informação e gerimos o nosso dia. Usando os recursos certos, seria possível criar uma aplicação que pudesse projetar os serviços de um PT para outros patamares. Para isso, foi definido um objetivo bem claro para a aplicação, dado que inicialmente existiram várias ideias de como se poderia auxiliar o papel do PT no acompanhamento, que, no entanto, acabavam por não cumprir o seu propósito. Por isso, propôs-se criar um algoritmo de gestão de exercícios baseado no cardiofrequencímetro do Apple Watch, selecionando dinamicamente os mesmos e enviando toda a informação necessária para o iPhone do PT.

Dado que, profissionalmente, já havia conhecimento de iOS, que a ITSector dispunha de dispositivos Apple Watch para testes e que a base da aplicação PTrainer estaria desenvolvida em Objective-C, foi decidido continuar com esta tecnologia e desenvolver este novo projeto recorrendo à mesma. Ainda assim, houve a oportunidade de solidificar alguns conhecimentos sobre o **WatchConnectivity** e o **HealthKit**, referidos, respetivamente, nos subcapítulos 5.10.3 e 5.10.2. No entanto, houve a preocupação de tornar todos os módulos genéricos, de forma que a possível passagem para Swift seja o mais simples e rápido.

No capítulo 5, foi detalhada toda a **arquitetura da aplicação**, a sua estrutura e componentes, bem como o seu funcionamento, recorrendo-se a vários diagramas que explicam de forma visual todos os módulos existentes. Houve ainda a possibilidade de implementar uma interface gráfica (subcapítulo 5.4) onde se deu destaque às cores base do PTrainer, servindo de base para uma possível iteração.

Aquando do desenvolvimento da aplicação, foi necessário investigar e estudar estas novas tecnologias, de forma a tirar partido das mesmas. No entanto, surgiram várias dificuldades técnicas, nomeadamente em manter a fiabilidade da aplicação num **ambiente assíncrono**, onde o Apple Watch envia e recebe mensagens, tal como o iPhone. Este mecanismo foi bastante desafiante, uma vez que durante um caso real, inúmeros problemas poderiam ocorrer e deteriorar a experiência de utilização da aplicação.

O uso do Apple Watch para testes reais também foi uma fonte de frustração, uma vez que este dispositivo foi utilizado para testes em muitos outros projetos na ITSector, tornando-os mais complexos do que o expectável. Além disso, como se tratava de um dispositivo da empresa, não foi possível utilizá-lo num ginásio, em paralelo com a aplicação.

Houve ainda alguns contratemplos em relação à base de dados e ao *Middleware*, que iriam ser utilizados e atualizados como serviços externos da aplicação. No entanto, durante o desenvolvimento desta dissertação e do projeto, houve uma **mudança de entidade empregadora** o que, como consequência, invalidou os acessos a estes componentes. Assim, apenas a aplicação, que fora desenvolvida como um módulo externo, poderia ser integrada em qualquer ambiente tecnológico.

É importante referir a prontidão dos orientadores do projeto, uma vez que forneceram todo o material necessário para detalhar os conteúdos teóricos, auxiliando ainda na componente técnica do desenvolvimento da aplicação móvel.

De forma a **avaliar a solução (5.7)**, foi proposta a criação de um questionário direcionado para cada PT, com o objetivo de avaliar a aplicação, descrever possíveis melhorias no acompanhamento dos seus clientes e no aumento da adesão dos mesmos. Foi também proposta a criação de notificações na aplicação, em forma de *pop-up*, a questionar de uma forma geral a experiência de utilização.

Estas hipóteses seriam uma forma de perceber se a solução está realmente a atingir o que é proposto. No entanto, o que poderia avaliar de forma mais concreta o objetivo deste projeto seria **uma fase de utilização da aplicação durante 5 a 10 meses** por parte de um PT. Desta

forma, o PT colocaria à prova a aplicação e daria feedback da mesma. Tal não foi possível, uma vez que nenhum ginásio contactado mostrou disponibilidade para utilizar a aplicação, tendo alguns deles protocolos estabelecidos com outras aplicações.

Seria interessante perceber até que ponto esta solução ia de encontro com o pretendido e de que forma se poderia criar funcionalidades em conjunto com um PT.

Houve também a oportunidade de estudar e definir os conceitos mais teóricos do negócio, nomeadamente a definição de **Valor (3.3)**, a definição do **Modelo Canvas (3.6)**, do **New Concept Development Model (3.2)**, do **Modelo AHP (3.8)** e a **Rede de Valor (3.7)**, o que promoveu um maior conhecimento sobre estes aspetos, tentando perceber o problema, encontrar valor tendo em conta o mercado, as oportunidades e as ameaças existentes. É importante estar à vontade com todos estes conceitos, uma vez que a nossa área de *expertise* é o desenvolvimento de software e com estes complementos, são fortalecidas as nossas competências na área de engenharia.

Outra componente de não menor importância é a do *fitness* no subcapítulo 5.5 e de toda a área de exercício-físico que foi necessário estudar para criar valor e perceber como seria possível auxiliar e fornecer informação útil para o PT, esta componente foi reforçada com a publicação do documento científico com ajuda do orientador e de um professor da área de Desporto.

Em suma, foi extremamente gratificante estudar uma possibilidade de negócio dentro da área do *fitness* com a colaboração de professores, colegas de trabalho e PT's. Não foi fácil conciliar a vida profissional com a elaboração desta dissertação e do projeto que esta documenta, mas acabou por se tornar mais simples com a ajuda de várias pessoas já referenciadas.

8 Referências

- Adidas. (2016). Adidas Annual Report. *Adidas*, 1.
- AGAP. (2016). Sumário Executivo. *Barometro*, 1.
- Allee, V. (2012). Value Network Analysis. *ValueNet Works*, 4.
- Alto, P., Shangai, & Reading, S. a. (14 de Novembro de 2017). *Apple retakes the lead in the wearable band market in Q3 2017*. Obtido de Canalsys: <https://www.canalys.com/newsroom/media-alert-apple-retakes-lead-wearable-band-market-q3-2017>
- Amazfit. (30 de Agosto de 2016). *What features do you want to see with the Amazfit app?* Obtido de Amazfit Support: <https://support.amazfit.com/hc/en-us/community/posts/214013728-What-features-do-you-want-to-see-with-the-Amazfit-app->
- AndroidPit. (13 de 01 de 2019). *AndroidPit*. Obtido de AndroidPit: <https://www.androidpit.com/>
- Apple. (s.d.). Obtido de Apple Watch: <https://www.apple.com/pt/apple-watch-series-3/>
- Apple. (s.d.). Obtido de WatchOS: <https://www.apple.com/pt/watchos/>
- Apple. (2012). *Delegates and Data Sources*. Obtido de Apple Documentation Archive: <https://developer.apple.com/library/archive/documentation/General/Conceptual/CocoaEncyclopedia/DelegatesandDataSources/DelegatesandDataSources.html>
- Apple. (17 de Setembro de 2014). *About Objective-C*. Obtido de Documentation Archive: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>
- Apple. (17 de Setembro de 2014). *Defining Classes*. Obtido de Documentation Archive: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/DefiningClasses/DefiningClasses.html#/apple_ref/doc/uid/TP40011210-CH3-SW5
- Apple. (17 de Setembro de 2014). *Encapsulating Data*. Obtido de Documentation Archive: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/EncapsulatingData/EncapsulatingData.html>
- Apple. (12 de Dezembro de 2016). *The Watch App Architecture*. Obtido de Apple Developer: <https://developer.apple.com/library/content/documentation/General/Conceptual/WatchKitProgrammingGuide/DesigningaWatchKitApp.html>

Apple. (14 de Dezembro de 2017). *Nike Training Club*. Obtido de Itunes:
<https://itunes.apple.com/us/app/nike-training-club/id301521403?mt=8>

Apple. (12 de Dezembro de 2017). *PTrainer*. Obtido de Itunes Apple:
<https://itunes.apple.com/pt/app/ptrainer/id1027842134?mt=8>

Apple. (28 de Dezembro de 2017). *Runkeeper*. Obtido de Itunes:
<https://itunes.apple.com/us/app/runkeeper-gps-running-tracker/id300235330?mt=8>

Apple. (15 de Dezembro de 2017). *Zova*. Obtido de Itunes:
<https://itunes.apple.com/au/app/zova-workout-fitness-trainer/id919000809?mt=8>

Apple. (2018). *ActivateSession*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsession/1615655-activate>

Apple. (2018). *App Store*. Obtido de Apple: <https://www.apple.com/pt/ios/app-store/>

Apple. (2018). *applicationDidBecomeActive*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchkit/wkextensiondelegate/1628185-applicationdidbecomeactive>

Apple. (2018). *applicationDidEnterBackground*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchkit/wkextensiondelegate/1650865-applicationdidenterbackground>

Apple. (2018). *applicationDidFinishLaunching*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchkit/wkextensiondelegate/1628241-applicationdidfinishlaunching>

Apple. (2018). *applicationWillEnterForeground*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchkit/wkextensiondelegate/1650868-applicationwillenterforeground>

Apple. (2018). *applicationWillResignActive*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchkit/wkextensiondelegate/1628206-applicationwillresignactive>

Apple. (2018). *ARKit*. Obtido de Apple Developer: <https://developer.apple.com/arkit/>

Apple. (2018). *Close Your Rings*. Obtido de Apple: <https://www.apple.com/watch/close-your-rings/>

Apple. (2018). *DefaultSession*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsession/1615673-defaultsession>

Apple. (2018). *HealthKit*. Obtido de Apple Developer: <https://developer.apple.com/healthkit/>

- Apple. (2018). *HealthKit*. Obtido de Human Interface Guidelines: <https://developer.apple.com/design/human-interface-guidelines/healthkit/overview/introduction/>
- Apple. (2018). *HKAnchoredObjectQuery*. Obtido de Apple Developer: <https://developer.apple.com/documentation/healthkit/hkanchoredobjectquery>
- Apple. (2018). *HKHealthStore*. Obtido de Apple Developer: <https://developer.apple.com/documentation/healthkit/hkhealthstore>
- Apple. (2018). *HKQuery*. Obtido de Apple Developer: <https://developer.apple.com/documentation/healthkit/hkquery?language=objc>
- Apple. (2018). *HKUnit*. Obtido de Apple Developer: <https://developer.apple.com/documentation/healthkit/hkunit>
- Apple. (2018). *HKWorkoutConfiguration*. Obtido de Apple Developer: <https://developer.apple.com/documentation/healthkit/hkworkoutconfiguration>
- Apple. (2018). *HKWorkoutSession*. Obtido de Apple Developer: <https://developer.apple.com/documentation/healthkit/hkworkoutsession>
- Apple. (2018). *HKWorkoutSessionDelegate*. Obtido de Apple Developer: <https://developer.apple.com/documentation/healthkit/hkworkoutsessiondelegate>
- Apple. (21 de Agosto de 2018). *Human Interface Guidelines*. Obtido de Human Interface Guidelines: <https://developer.apple.com/design/human-interface-guidelines/>
- Apple. (2018). *isSupported*. Obtido de Apple Developer: <https://developer.apple.com/documentation/watchconnectivity/wcsession/1615647-issupported>
- Apple. (2018). *isWatchAppInstalled*. Obtido de Apple Developer: <https://developer.apple.com/documentation/watchconnectivity/wcsession/1615623-iswatchappinstalled>
- Apple. (2018). *Lightning to USB Cable* . Obtido de Apple Accessories: <https://www.apple.com/shop/product/MQUE2AM/A/lightning-to-usb-cable-1-m>
- Apple. (4 de Abril de 2018). *Nike+ Run Club*. Obtido de iTunes: <https://itunes.apple.com/us/app/nike-run-club/id387771637?mt=8>
- Apple. (2018). *NSNotificationCenter*. Obtido de Apple Developer: <https://developer.apple.com/documentation/foundation/nsnotificationcenter?language=objc>
- Apple. (2018). *NSPredicate*. Obtido de Apple Developer: <https://developer.apple.com/documentation/foundation/nspredicate>

- Apple. (2018). *reachable*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsession/1615683-reachable>
- Apple. (2018). *sendMessage*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsession/1615687-sendmessage>
- Apple. (2018). *session(_:didReceiveMessage:replyHandler:)*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsessiondelegate/1615677-session>
- Apple. (2018). *session:activationDidCompleteWithState:error:*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsessiondelegate/1615679-session?language=objc>
- Apple. (2018). *sessionDidBecomeInactive*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsessiondelegate/1615684-sessiondidbecomeinactive>
- Apple. (2018). *sessionDidDeactivate*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsessiondelegate/1615670-sessiondiddeactivate>
- Apple. (2018). *sessionReachabilityDidChange*. Obtido de Apple Developer:
<https://developer.apple.com/documentation/watchconnectivity/wcsessiondelegate/1615662-sessionreachabilitydidchange>
- Apple. (4 de Maio de 2018). *Set up your Apple Watch*. Obtido de Apple Support:
<https://support.apple.com/en-us/HT204505>
- Apple. (2018). *Sharing Data*. Obtido de Apple Documentation Archive:
https://developer.apple.com/library/archive/documentation/General/Conceptual/WatchKitProgrammingGuide/SharingData.html#//apple_ref/doc/uid/TP40014969-CH29-SW1
- Apple. (2018). *Storyboard*. Obtido de Developer Documentation:
<https://developer.apple.com/documentation/uikit/uiviewController/1621399-storyboard>
- Apple. (2018). *Swift 4*. Obtido de <https://developer.apple.com/swift/>
- Apple. (2018). *The Watch App Architecture*. Obtido de Apple Documentation Archive:
https://developer.apple.com/library/archive/documentation/General/Conceptual/WatchKitProgrammingGuide/DesigningaWatchKitApp.html#//apple_ref/doc/uid/TP40014969-CH3-SW1

Apple. (8 de Fevereiro de 2018). *Utilizar apps no Apple Watch*. Obtido de Apple Suporte: <https://support.apple.com/pt-pt/HT204784>

Apple. (21 de Agosto de 2018). *Watch Connectivity*. Obtido de Apple Developer Documentation: <https://developer.apple.com/documentation/watchconnectivity>

Apple. (2018). *WatchConnectivity*. Obtido de Apple Developer: <https://developer.apple.com/documentation/watchconnectivity>

Apple. (2018). *WCSession*. Obtido de Apple Developer: <https://developer.apple.com/documentation/watchconnectivity/wcsession>

Apple. (2018). *WCSessionDelegate*. Obtido de Apple Developer: <https://developer.apple.com/documentation/watchconnectivity/wcsessiondelegate>

Apple. (2018). *Workouts*. Obtido de Apple Developer: <https://developer.apple.com/documentation/sirikit/workouts>

Apple. (2018). *Workouts and Activity Rings*. Obtido de Apple Developer: https://developer.apple.com/documentation/healthkit/workouts_and_activity_rings

Apple. (19 de Setembro de 2019). Obtido de Breathe App: <https://support.apple.com/en-us/HT206999>

Apple. (29 de Setembro de 2019). Obtido de Efetuar um ECG com a app ECG no Apple Watch Series 4 ou posterior: <https://support.apple.com/pt-pt/HT208955>

Apple. (2019). Obtido de WatchOS 6: <https://www.apple.com/pt/watchos/watchos-6/>

Apple. (2019). *Apple Watch Series 5*. Obtido de <https://www.apple.com/pt/apple-watch-series-5/>

Apple. (2019). *Health*. Obtido de Apple: <https://www.apple.com/ios/health/>

Apple. (2019). *iPhone 11 Pro*. Obtido de <https://www.apple.com/pt/iphone/>

Apple. (2019). *Xcode 11*. Obtido de Apple Developer: <https://developer.apple.com/xcode/>

Apple. (s.d.). *NSObject*. Obtido de Developer Documentation: <https://developer.apple.com/documentation/objectivec/nsobject>

Bain, M. (23 de Março de 2017). *Here's what happens to the athletic wear industry when China starts going to the gym*. Obtido de Quartz: <https://qz.com/939224/heres-what-happens-to-the-athletic-wear-industry-when-china-starts-going-to-the-gym/>

Balance, N. (16 de Junho de 2011). *New Balance Announces Strong Growth In Sales At Licensed Stores*. Obtido de New Balance: https://www.newbalance.com/press-releases/id/press_2011_growth_sales_licensed_stores.html?fbclid=IwAR01YTEbfRuBkXG1IUYMLSn1v3hGryBYM_07MjKK0csNhf1vm8v1IHPyvFY

- Campos, M. (22 de Novembro de 2017). *Heart rate variability: A new way to track well-being*. Obtido de Harvard Health Publishing: <https://www.health.harvard.edu/blog/heart-rate-variability-new-way-track-well-2017112212789>
- CardioSmart. (2015). Exercise: Measuring Intensity. *American College of Cardiology*.
- Ching, C. (16 de Janeiro de 2018). *Xcode Tutorial for Beginners*. Obtido de Code with Chris: <https://codewithchris.com/xcode-tutorial/>
- Coelho, I. (1 de Setembro de 2016). Obtido de Pplware: <https://pplware.sapo.pt/analises-2/analise-samsung-gear-fit2/>
- Coelho, I. (6 de Março de 2017). Obtido de Pplware: <https://pplware.sapo.pt/analises-2/analise-ao-smartwatch-samsung-gear-s3/>
- Duarte, A. (29 de Agosto de 2017). Obtido de Economista: <http://www.economista.pt/artigo/apps-para-viciados-em-exercicio-fisico/>
- EINOV. (2017). *Pensar o Negócio - EINOV*. ISEP.
- Faulkner, C. (20 de Maio de 2018). *Samsung Gear Fit 2*. Obtido de Techradar: <https://www.techradar.com/reviews/wearables/samsung-gear-fit-2-1322606/review>
- Ferreira, A., Coelho, J., & Nogueira, N. (30 de Agosto de 2018). Wearables with Heart Rate Monitors and Dynamic Workout Plans. *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1-5.
- Fibit. (28 de Abril de 2018). *Charge HR First Steps*. Obtido de Fibit: <https://www.fitbit.com/eu/chargehr/chargehr-101>
- Finance, B. (2019). *Nike Just Did It Again as World's Most Valuable Apparel Brand*. Obtido de Brand Finance: <https://brandfinance.com/news/press-releases/nike-just-did-it-again-as-worlds-most-valuable-apparel-brand/>
- Gusgård, O. (2018 de Maio de 14). Application development for the Apple Watch. p. 10.
- Heuer, T. (s.d.). Obtido de TAG HEUER CONNECTED MODULAR: <https://www.tagheuer.com/pt/relogios/tag-heuer-connected>
- Ikeda, A., & Oliveira, T. (2005). O CONCEITO DE VALOR PARA O CLIENTE: DEFINIÇÕES E IMPLICAÇÕES GERENCIAIS EM MARKETING. *REAd*, 1.
- Jorge, & Simões, V. (2010). Os determinantes do valor da marca : um estudo de caso na indústria farmacêutica.
- Klotzbach, C., & Kesiraju, L. (27 de Dezembro de 2017). Obtido de Flurry Analytics Blog: <http://flurrymobile.tumblr.com/#.U6LhLMypoll>

- Koen, P. (2018). *What is the New Concept Development (NCD) model?* Obtido de Frontend Innovation: <http://www.frontendinnovation.com/fei>
- Koen, P., Bertels, H., & Kleinschmidt, E. (2014). Managing the Front End of Innovation—Part II. *RESEARCH-ON-RESEARCH*, 25.
- Krasner, G., & Pope, S. (1988). A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. *ParcPlace Systems*.
- Laurinavicius, T. (16 de Maio de 2016). *This Is How Gyms Should Be Taking Advantage Of Technology*. Obtido de Forbes: <https://www.forbes.com/sites/tomaslaurinavicius/2016/05/16/gyms-technology/#2e3825946c93>
- Lister, C., West, J., Cannon, B., Sax, T., & Brodegard, D. (2014). Just a Fad? Gamification in Health and Fitness Apps. *US National Library of Medicine*, 10.
- Lora, K. (12 de Setembro de 2017). Obtido de CNBC: <https://www.cnbc.com/2017/09/12/apple-watch-is-now-the-number-one-watch-in-the-world.html>
- Mallawaarachchi, V. (4 de Setembro de 2017). *10 Common Software Architectural Patterns in a nutshell*. Obtido de Towards Data Science: <https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013>
- Martins, C. (2001). Valor para o Cliente: Uma análise no ramo de Agronegócios. 32.
- Mi. (s.d.). Obtido de Mi Band: <https://www.mi.com/global/mi-smart-band-4>
- Microsoft. (10 de Fevereiro de 2012). *The MVVM Pattern*. Obtido de Microsoft Developer Network: <https://msdn.microsoft.com/en-us/library/hh848246.aspx>
- Microsoft. (10 de Outubro de 2017). *Implementing the Model-View-ViewModel Pattern*. Obtido de Microsoft Developer Network: <https://msdn.microsoft.com/en-us/library/ff798384.aspx>
- Mocenni, C. (s.d.). The Analytic Hierarchy Process. *Department of Information Engineering and Mathematics - University of Siena*, 1.
- Nike. (2017). NIKE, INC. REPORTS FISCAL 2017 THIRD QUARTER RESULTS. *Nike*, 1.
- Nunes, E. (3 de Outubro de 2015). *Prozis: Da ideia ao topo do mercado em nove anos*. Obtido de Dinheiro Vivo: <https://www.dinheirovivo.pt/fazedores/prozis-da-ideia-ao-topo-do-mercado-em-nove-anos/>
- ONAP. (23 de Abril de 2018). *Síntese de Resultados*. Obtido de Observatório da Nutrição e Alimentação: <http://onap.premivalor.pt/onap/sintese-dos-resultados/>

- Padilha, A. (2004). Usabilidade na Web: uma Proposta de Questionário para Avaliação do Grau de Satisfação de Usuários do Comércio Eletrônico. *UNIVERSIDADE FEDERAL DE SANTA CATARINA*, 46.
- Pereira, D. (8 de Julho de 2016). *O que é o Business Model Canvas*. Obtido de O analista de modelos de negócio: <https://analistamodelosdenegocios.com.br/o-que-e-o-business-model-canvas/>
- Publicidade, M. e. (6 de Junho de 20016). *Meios e Publicidade*. Obtido de <http://www.meiosepublicidade.pt/2016/06/e-este-o-retrato-do-sector-de-ginasios-em-portugal/>
- Quinn, E. (17 de Fevereiro de 2018). *VO2 Max and How It Is Measured in Athletes*. Obtido de Verywell Fit: <https://www.verywellfit.com/what-is-vo2-max-3120097>
- Ridgers, N. (2016). Feasibility and Effectiveness of Using Wearable Activity Trackers in Youth: A Systematic Review. *JMIR Mhealth Uhealth*, 1.
- Sadun, E. (2011). *iOS 4 Developer's Cookbook, The: Core Concepts and Essential Recipes for iOS Programmers*. Developer's Library.
- Salutes, B. (14 de Julho de 2017). Obtido de AndroidPit: <https://www.androidpit.com.br/melhores-pulseiras-para-fitness>
- Samsung. (s.d.). Obtido de Samsung Gear S3 Frontier: <http://www.samsung.com/pt/gear-s3/>
- Samsung. (s.d.). Obtido de Samsung Gear Fit 2: <http://www.samsung.com/pt/wearables/gear-fit-2-r360/SM-R3600DAATPH/>
- Samsung. (20 de Maio de 2018). *Gear Fit2 Pro*. Obtido de Samsung: <http://www.samsung.com/pt/wearables/gear-fit2-pro/SM-R365NZKATPH/>
- Samsung. (s.d.). *Samsung Gear S3 Apps list*. Obtido de Samsung : <http://www.samsung.com/global/galaxy/gear-s3/experience/>
- Shcherbina, A., Mattsson, M., Waggott, D., Salisbury, H., Christle, J., Hastie, T., . . . Ashley, E. (2017). Accuracy in Wrist-Worn, Sensor-Based Measurements of Heart Rate and Energy Expenditure in a Diverse Cohort. *Journal of Personalized Medicine*, 3.
- Sjøgaard, G., Christensen, J., Justesen, J., Murray, M., Dalager, T., Fredslund, G., & Sjøgaard, K. (2016). Exercise is more than medicine: The working age population's well-being and productivity. *Journal of Sport and Health Science* 5, 5.
- Smith, D. (27 de Agosto de 2018). *iOS Version Stats*. Obtido de DavidSmith: <https://david-smith.org/iosversionstats/>
- Sony. (22 de Abril de 2018). *Smartband 2*. Obtido de Sony Mobile: <https://www.sonymobile.com/pt/products/smart-products/smartband-2/>

- Spotify. (30 de Outubro de 2018). *Samsung wearables*. Obtido de Spotify Support: https://support.spotify.com/us/listen_everywhere/smart_watches/samsung-fit-gear-2/
- Stahl, S., An, H.-S., Dinkel, D., Noble, J., & Lee, J.-M. (2016). How accurate are the wrist-based heart rate monitors during walking and running activities? Are they accurate enough? *BMJ Open Sport & Exercise Medicine*, 5-7.
- Statista. (1 de Junho de 2017). *Cumulative number of apps downloaded from the Apple App Store from July 2008 to June 2017*. Obtido de Statista: <https://www.statista.com/statistics/263794/number-of-downloads-from-the-apple-app-store/>
- Statista. (2018). *Unit sales of the Apple iPhone worldwide from 2007 to 2017 (in millions)*. Obtido de The Statistics Portal: <https://www.statista.com/statistics/276306/global-apple-iphone-sales-since-fiscal-year-2007/>
- Tanaka, H., Monahan, K., & Seals, D. (13 de Setembro de 2000). Age-Predicted Maximal Heart Rate Revisited. *Journal of the American College of Cardiology*, pp. 154-156.
- Technology, S. I. (2017). *Organization FEI*. Obtido de Stevens Institute of Technology: <https://web.stevens.edu/cce/NEW/optFEI.htm>
- Teza, P., Miguez, V., Fernandes, R., & Dandolini, G. (2013). Direcionadores do processo de inovação: o papel da estratégia, liderança e cultura. *ResearchGate*, 1-10.
- TIOBE. (Agosto de 2018). *TIOBE Index for August 2018*. Obtido de TIOBE: <https://www.tiobe.com/tiobe-index/>
- Trew, J. (23 de Junho de 2014). Obtido de Engadget: <https://www.engadget.com/2014/06/23/sony-smartband-review/>
- T-Test. (24 de Fevereiro de 2018). Obtido de Investopedia: <https://www.investopedia.com/terms/t/t-test.asp>
- Usirono, C. (2015). Engenharia e Análise de Valor. *EUAX*, 10.
- W3sDesign. (2018). *Singleton*. Obtido de W3sDesign: <http://w3sdesign.com/?gr=c05&ugr=proble>
- Wayner, P. (5 de Fevereiro de 2018). *The top 5 software architecture patterns: How to make the right choice*. Obtido de TechBeacon: <https://techbeacon.com/top-5-software-architecture-patterns-how-make-right-choice>
- Wewege, M., Berg, R. v., Ward, R. E., & Keech, A. (12 de Fevereiro de 2017). The effects of high-intensity interval training vs. moderate-intensity continuous training on body composition in overweight and obese adults: a sytematic review and meta-analysis. *Etiology and Pathophysiology*, pp. 636-639.

Wikipedia. (7 de Agosto de 2019). *iPhone*. Obtido de Wikipedia:
<https://en.wikipedia.org/wiki/IPhone>

Zeis, A. (13 de Janeiro de 2014). *The best smartwatches from CES 2014*. Obtido de Smartwatchfans:
<https://web.archive.org/web/20140330184434/http://www.smartwatchfans.com/best-smartwatches-ces-2014>

Zhang, H., Tong, T. K., Qiu, W., Zhang, X., Zhou, S., Liu, Y., & He, a. Y. (1 de Janeiro de 2017). Comparable Effects of High-Intensity Interval Training and Prolonged Continuous Exercise Training on Abdominal Visceral Fat Reduction in Obese Young Women. *Hindawi*, pp. 1-8.