



Sistema de apoio à decisão de treinos à equipa técnica

HUGO ANDRÉ RIBEIRO SOARES

Outubro de 2019

Sistema de apoio à decisão de treinos à equipa técnica

Hugo André Ribeiro Soares

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

Orientador: Professor Doutor António Constantino Martins

Coorientador: Professor Paulo Matos

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, Outubro de 2019

Aos meus pais por me terem apoiado até aqui.

À minha namorada, Leonor Corte-Real, pelo apoio e ajuda incansáveis.

Resumo

Nos dias de hoje, em todas as modalidades desportivas, as equipas procuram criar todas as condições que permitam um melhor desempenho das mesmas. Inevitavelmente, a tecnologia tornou-se uma ferramenta preponderante para o aumento dessas condições. Sendo este um mercado em ascensão, especificamente no futebol, modalidade que ultimamente tem adotado cada vez mais as tecnologias para tornar o espetáculo mais atrativo, as equipas estão a investir cada vez mais em softwares que permitam ter vantagens em relação aos concorrentes.

Esta, é a necessidade que torna o projeto cada vez mais atrativo e inovador, dado que atualmente não existe no mercado softwares com as mesmas características.

O projeto tem como objetivo primário a criação de um sistema de classificação de exercícios de futebol por atributo, sendo este um sistema inovador, que irá permitir a todos os intervenientes interessados e que pertencem a uma equipa da modalidade, aumentar a performance desportiva. O sistema irá permitir catalogar os exercícios por atributos específicos, por exemplo, um exercício vai estar associado ao atributo agilidade. O desenvolvimento do mesmo terá em conta um tipo de abordagem de classificação baseada em aprendizagem, permitindo assim posteriormente a integração com um sistema de recomendação de treinos.

Em suma, este projeto irá consistir numa aplicação que auxiliará os treinadores, mais especificamente, com um método de classificação de exercícios permitirá ao sistema recomendar treinos com base no mesmo método.

Palavras-chave: Sistema de Recomendação, Futebol, Exercícios, Atributos chaves

Abstract

Nowadays, in every sports modality, teams try to create conditions that allows them to do a better performance. Inevitably, technology became an overweight tool to increase that conditions. Being this a rising market, specifically in football, modality that as adopted increasingly their technologies lately to make the show more attractive, teams are investing more and more in software that allows them to have advantages over others.

The fact that it doesn't exist any other software with the same characteristics on the market nowadays is what makes this project more attractive and innovator.

This project has as his main and primer goal the creation of a classification football exercises system by attribute. This innovator factor is what allows, the interested stakeholders and the people that is on a team of this modality, to increase their performance.

The system will allow the categorization of the exercises by specific attributes, for example, one exercise will be associated to the agility attribute. Its development will take into account a type of learning-based grading approach, thus enabling further integration with a training recommendation system.

In conclusion, this project is based on an application that helps coaches, more specifically, the classification method classifies exercises which helps the system with the training recommendation.

Keywords: Recommendation System, football, exercises, key attributes.

Agradecimentos

Em primeiro lugar agradeço aos meus orientadores e coorientadores, Professor Constantino Martins e Professor Paulo Matos, pela ajuda e conselhos.

Aos meus pais e à minha namorada, por sempre terem acreditado que este dia iria chegar.

Obrigado por tudo. Hugo Soraes

Índice

1	Introdução	22
1.1	Contexto	22
1.2	Problema.....	24
1.3	Objetivos.....	24
1.4	Resultados esperados	25
1.5	Estrutura do Documento	26
2	Estado da arte	29
2.1	Sistemas de classificação de informação	29
2.1.1	Classificação baseada em Aprendizagem	29
2.1.2	Métodos estatísticos de classificação	32
2.2	Sistemas de classificação de treino desportivo	34
2.3	Softwares existentes de apoio ao treinador.....	36
2.3.1	Metodologia de pesquisa	36
2.3.2	TaticalBorders soccer	37
2.3.3	Coach-helper	38
2.3.4	MyCoach.....	39
2.3.5	Soccer-trainer	40
2.3.6	Newsoccerdrills	41
2.3.7	Comparação entre funcionalidades	42
3	Análise de Valor	45
3.1.1	Novo Modelo de Desenvolvimento de Conceito (NCD)	46
3.1.2	Valor, Valor Percecionado e Valor para o cliente	50
3.1.3	Proposta de Valor	52
3.1.4	Modelo Canvas	53
4	Análise e Design	56
4.1	Visão da Solução	56
4.2	Análise e Conceção	57
4.2.1	Stakeholders.....	58
4.2.2	Atores	58
4.2.3	Modelo de Domínio	59
4.2.4	Requisitos funcionais	62
4.2.5	Requisitos não funcionais	64
4.3	Arquitetura da Solução	66
4.3.1	MVC (Model-View-Controller)	67
4.3.2	Vista Lógica.....	69
4.3.3	Vista de Implantação	74
4.3.4	Vista de Implementação	74
4.3.5	Vista de Dados.....	76

4.3.6	Vista de Casos de Uso (Cenários)	76
5	Implementação	80
5.1	Aplicação SmartCoachManager	80
5.1.1	Domain	81
5.1.2	Application	82
5.1.3	Data	82
5.1.4	Presentation.....	83
5.1.5	Evidências de boas práticas e do uso de padrões de <i>software</i>	84
5.2	Casos de Uso	85
5.2.1	Login/Registo na aplicação.....	86
5.2.2	Gestão de Atributos	87
5.2.3	Criação de exercícios e classificação de atributos	88
5.2.4	Editar/Visualizar/Eliminar exercícios.....	91
5.2.5	Criação manual de treinos.....	93
5.2.6	Criação automática de treinos com base no algoritmo de recomendação	96
5.2.7	Editar/Eliminar/Visualizar detalhes do treino	99
5.2.8	Visualizar calendário com agendamento dos treinos	101
5.2.9	Dashboard estatístico	102
5.3	Sistema de Classificação e Recomendação.....	103
6	Avaliação	110
6.1	Grandezas	110
6.2	Metodologia	111
6.2.1	Modelo V	112
6.3	Implementação dos testes	113
6.3.1	Testes Unitários.....	114
6.3.2	Testes de Integração	114
6.3.3	Testes de Sistema	115
6.3.4	Testes de aceitação	117
6.4	Avaliação de usabilidade	118
6.4.1	Metodologia para a sessão de avaliação	118
6.4.2	Resultados e interpretação de resultados.....	119
6.4.3	Apreciações finais	124
7	Conclusões	126
7.1	Resumo	126
7.2	Objetivos Alcançados	127
7.3	Limitações e Trabalho Futuro.....	129
	Referências	131
	Anexos	134
	Anexo 1 - Modelo de dados ER	134
	Anexo 2 - Guião de testes	135

Anexo 3 - Questionário.....147

Lista de Figuras

Figura 1 Estudo sobre o aumento da utilização das apps de desporto	23
Figura 2 Tactical Borders Soccer	38
Figura 3 Coach-Helper	39
Figura 4 MyCoach Football	40
Figura 5 Soccer-Trainer Online	41
Figura 6 New Soccer Drills	42
Figura 7 O modelo New Concept Development (NCD)	46
Figura 8 Modelo Canvas.....	54
Figura 9 Diagrama de processo de Classificação	57
Figura 10 Modelo de Arquitetura 4+1	66
Figura 11 - Padrão Model-View-Controller.....	69
Figura 12 Diagrama de Componentes da Arquitetura Geral.....	70
Figura 13 Diagrama de Componentes da Arquitetura Backend.....	71
Figura 14 Diagrama de Componentes da Arquitetura Frontend.....	73
Figura 15 - Vista de Implantação	74
Figura 16 – Vista de Implementação	75
Figura 17 - Organização do projeto	81
Figura 18 - Vista de módulos da componente Domain	81
Figura 19 - Vista de módulos da componente Application.....	82
Figura 20 - Vista de módulos da componente Data	83
Figura 21 - Vista de módulos da componente Presentation	83
Figura 22 - Implementação do padrão Repositório.....	85
Figura 23 – Registo na aplicação.....	86
Figura 24 – Iniciar sessão na aplicação	86
Figura 25 – Diagrama de Sequência: Gestão de atributos	87
Figura 26 – Gestão de atributos.....	88
Figura 27 – Diagrama de Sequência: Criar Exercício e Classificar	89
Figura 28 - Criar exercício	90
Figura 29 - Selecionar atributos.....	91
Figura 30 - Classificar os atributos	91
Figura 31 - Visualizar detalhes do treino	93
Figura 32 - Diagrama de Sequência: Criar treino manual.....	94
Figura 33 - Criar o treino manual.....	95
Figura 34 - Selecionar exercícios.....	95
Figura 35 - Treino com exercícios	96
Figura 36 - Diagrama de Sequência: Criar treino com recomendação.....	97
Figura 37 - Obter recomendação.....	98
Figura 38 - Criar treino com recomendação.....	98
Figura 39 - Treino com exercícios recomendados	99
Figura 40 - Visualizar detalhes do treino	101

Figura 41 - Calendário desportivo	102
Figura 42 - Dashboard Estatístico	103
Figura 43 - Modelo V	113
Figura 44 - Organização dos testes da aplicação "SmartCoachManager"	113
Figura 45 - Teste Integração GetRecomendation	115
Figura 46 – Gráfico monitorização do tempo de resposta do algoritmo	116
Figura 47 - Escala Likert.....	119
Figura 48 - Cargos referentes à amostra.....	120
Figura 49 – Gráfico de opiniões “Design e usabilidade”	121
Figura 50 - Gráfico de opiniões "SmartCoachManager"	122
Figura 51 - Gráfico de opiniões Utilidade da aplicação.....	123
Figura 52 - Gráfico- Classificação da aplicação	123
Figura 53 - Gráfico de opiniões Recomendação da aplicação	124

Lista de Tabelas

Tabela 1 Classificação dos exercícios de treino [4].	34
Tabela 2 Classificação dos exercícios de treino em função da identidade [4].	36
Tabela 3 Tabela Comparativa das funcionalidades.	42
Tabela 4 Benefícios e Sacrifícios para o cliente	52
Tabela 5 – Especificação do caso de uso “UNR1: Registrar na aplicação”.	76
Tabela 6 - Especificação do caso de uso “UR3: Criar novo exercício”.	77
Tabela 7 - Especificação do caso de uso “UR5: Criar treino manual”.	77
Tabela 8 - Especificação do caso de uso “UR6 - Criar novo treino com recomendação”.	78
Tabela 9 - Teste de sistema	116
Tabela 10 - Teste de aceitação	117
Tabela 11 – Legenda do gráfico “Design e Usabilidade”	120
Tabela 12 - Legenda do gráfico "SmartCoachManager"	121
Tabela 13 - Sugestões de melhorias e comentários dos inquiridos	124
Tabela 14 - Objetivos propostos e estado de conclusão	128

Lista de Códigos

Código 1 - Model Atributo	88
Código 2 - Exemplo View Detalhes do exercício	92
Código 3 - Exemplo View Detalhes do treino	100
Código 4 - Model ExercicioTreino	104
Código 5 - Model AtributoAssigment	104
Código 6 - Model Recomendacao	105
Código 7 - Model AtributoRecomendacao	105
Código 8 – Serviço de comunicação com api externa	106
Código 9 - Model Treino	106
Código 10 - Model ExercicioAssigment	107
Código 11 - Algoritmo de Recomendação	108
Código 12 - Teste Unitário	114

Lista de Fórmulas

Fórmula 1 - Cálculo do valor de recomendação do exercício	108
---	-----

Acrónimos e Símbolos

Lista de Acrónimos

API	Application Programming Interface
CRUD	<i>Create, Read, Update, Delete</i>
FFE	<i>Fuzzy Front End</i>
FURPS	Functionality, Usability, Reliability, Performance, Supportability
GPS	<i>Global Positioning System</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
IPP	Instituto Politécnico do Porto
MVC	<i>Model-View-Controller</i>
NPD	<i>New Product Development</i>
NCD	<i>New Concept Development</i>
ORM	Object-Relational Mapper
SGBD	Sistema de Gestão de Base Dados
SQL	Structured Query Language
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>

1 Introdução

Neste capítulo introdutório, será apresentado o projeto da dissertação e o âmbito para qual este está a ser desenvolvido, a interpretação do problema, assim como os objetivos principais subjacentes. Por fim, é detalhada a estrutura do documento.

1.1 Contexto

As sociedades modernas são cada vez mais marcadas pelo lazer, assumindo o desporto neste quadro, uma das suas facetas mais expressivas. Dentro das diferentes modalidades, o futebol ocupa uma posição central, devido ao facto de ter aumentado a sua presença no nosso quotidiano a sua importância (económica, social e territorial) e como tal, tem registado um crescimento contínuo. Por estes motivos, pode ser considerado um dos fenómenos mais marcantes e controversos da civilização moderna, ocupando um lugar privilegiado na ocupação do tempo livre ¹.

Com o aumento da importância do desporto nas sociedades modernas, este, tornou-se um dos principais focos para o desenvolvimento de novas tecnologias. Sendo que, no desporto de alta competição, nomeadamente o Futebol que é a modalidade em que esta dissertação se insere, o uso destas tecnologias é cada vez mais importante, de forma a ter vantagens competitivas comparativamente com os adversários.

A tecnologia é sempre vista como uma mais valia desde que seja usada e implementada para fazer evoluir o espetáculo, como por exemplo através de mecanismos como a instalação de maior número de câmaras de televisão, ou para os árbitros poderem comunicar entre si em tempo real, numa tentativa de minimizar erros que possam prejudicar o espetáculo. Noutra vertente, a tecnologia está a ser cada vez mais usada por clubes, nomeadamente por treinadores, para se compreender como se pode

¹ Jacinto, Rui; Malta, Paula; DESPORTO, SOCIEDADE E TERRITORIO: O FUTEBOL NA REGIAO CENTRO.

aproveitar o maior rendimento dos jogadores e da equipa mediante a monitorização de vídeo e utilização de ferramentas como o cardiofrequencímetro ou gps, permitindo assim uma recolha de dados estatísticos relacionados com frequência cardíaca ou distância percorrida num jogo ².

O desenvolvimento de aplicações móveis relacionadas com o desporto e bem-estar tornou-se um dos principais focos por parte das empresas e investidores. Segundo o estudo realizado pela empresa Statista, as aplicações móveis de desporto e bem-estar estão em constante crescimento, sendo que em 2017 só na União Europeia quarenta e dois milhões de pessoas utilizavam este tipo de aplicações o que corresponde a um crescimento de 8,27%, prevendo-se que em 2020 haja um crescimento de 12,52% na utilização destas aplicações.

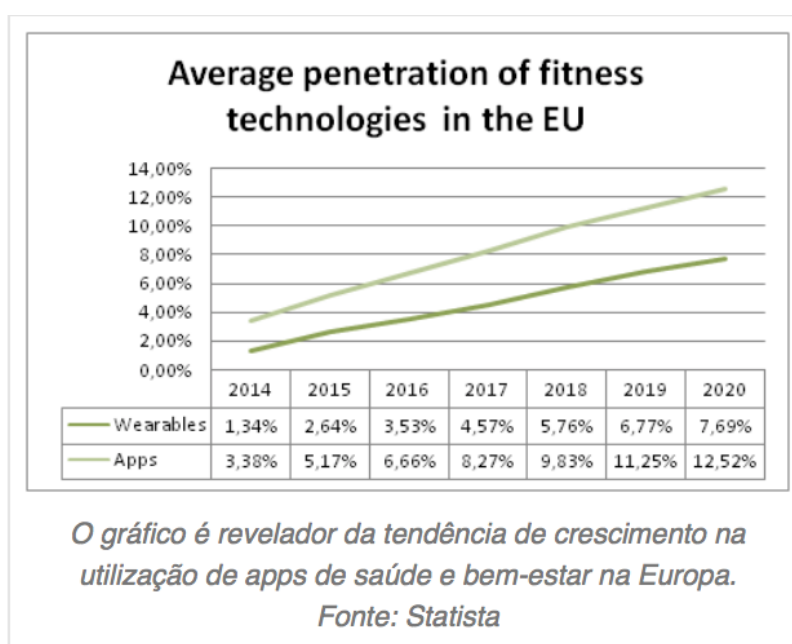


Figura 1 Estudo sobre o aumento da utilização das apps de desporto ³.

O presente projeto pretende assim inovar e causar impacto, através da combinação das novas tecnologias focadas em sistemas de classificação e filtragem ao ambiente

² Site: <https://www.easyapp.pt/blog/apps-de-saude-e-bem-estar/>

³ org

desportivo, de modo a melhorar a performance desportiva de atletas e treinadores de futebol.

1.2 Problema

A inteligência artificial, nomeadamente os sistemas de apoio à decisão estão a ser cada vez mais utilizados no apoio aos treinos no desporto, principalmente no futebol, com o objetivo de potenciar as decisões e os resultados das equipas técnicas.

É nesta vertente que este projeto pretende inovar e causar impacto, no uso da tecnologia e técnicas de inteligência artificial como suporte aos treinadores e staff técnico, para uma melhor análise sobre os seus atletas e sobre as potencialidades a melhorar nos treinos.

1.3 Objetivos

O objetivo desta dissertação é desenvolver um protótipo, que visa ajudar as equipas técnicas nas tomadas de decisões sobre que treinos aplicar à equipa ou a um determinado jogador no sentido de evoluir em grupo e individualmente. Para tal, pretende-se desenvolver um *software*, com um sistema de apoio à decisão, com o propósito de facilitar a interação entre membros do “Staff” técnico de um clube e a equipa.

Neste sentido, e para cumprir com o objetivo mencionado, foi definido um conjunto de tarefas específicas, tais como:

- Efetuar o levantamento e especificação de requisitos;
- Documentar as decisões tomadas sobre o design arquitetural;
- Desenvolvimento de protótipo que:
 - Permita a gestão de acessos;

- Forneça mecanismos de autenticação;
- Permita a gestão de atributos, exercícios e treinos;
- Contenha um sistema de classificação de treinos, que permita associar os atributos específicos ao treino e classificá-los;
- Permita a criação manual de treinos, dando a possibilidade ao utilizador de seleccionar os exercícios que pretende inserir no treino;
- Permita a criação de treinos através de um algoritmo de recomendação, que gera automaticamente um plano de treino com base no sistema de classificação de treinos;
- Possibilite a integração com um sistema de recomendação;
- Permita visualizar os treinos agendados;
- Forneça informações ao utilizador através de dados estatísticos;
- *User Interface (UI)* seja responsiva;
- Realizar testes:
 - Testes unitários;
 - Testes de integração;
 - Testes de sistema;
 - Testes de usabilidade através de uma equipa de futebol;

1.4 Resultados esperados

No final desta dissertação é esperado que se atinga os seguintes resultados:

1. **Solução de *software* desenvolvida e completamente funcional:** espera-se que no final do projeto o processo de gestão de atributos, exercícios e treinos esteja totalmente funcional. A plataforma será responsiva.
2. **Sistema de classificação de exercícios funcional:** espera-se que o processo de classificar atributos quanto ao exercício esteja totalmente funcional.
3. **Recomendação de treinos funcional:** espera-se que o processo de recomendação de treinos, com base na informação de classificação, esteja totalmente funcional.
4. **Realização de testes:** espera-se que a plataforma a desenvolver seja validada através de testes unitários, de integração, de sistema e de usabilidade.

1.5 Estrutura do Documento

A presente dissertação está estruturada em sete capítulos.

No primeiro capítulo, referente à Introdução, são definidos o contexto e o levantamento do problema, bem como, os objetivos a atingir no decorrer do projeto e os resultados esperados.

No segundo capítulo, primeiramente, é detalhado o estado da arte relativo a sistemas de classificação, e seguidamente, são identificados softwares existentes que auxiliem treinadores e jogadores de futebol a melhorar a sua atividade desportiva.

No terceiro capítulo é detalhada a análise de valor do projeto, esta tem como base o novo modelo de desenvolvimento de conceito, é explicado os cinco conceitos base do mesmo e aplicado ao projeto, é analisado com o auxílio de uma tabela os benefícios e sacrifícios do cliente e por fim, é apresentado o modelo de negócio com base no modelo CANVAS.

No quarto capítulo é apresentado o Design da solução, começando pela definição da visão da solução que dá resposta ao problema exposto, posteriormente, é realizada a

análise e conceção do projeto, contendo os requisitos funcionais e não funcionais, e por último, é exibida a arquitetura do sistema baseada no Modelo quatro mais um (4+1).

No quinto capítulo são apresentados os principais detalhes de implementação, com as devidas justificações.

No sexto capítulo é detalhado o tipo de avaliação feito ao projeto, são identificadas as métricas e grandezas que devem ser avaliadas em relação ao mesmo. De seguida, é apresentada e justificada a metodologia a utilizar na avaliação destas hipóteses, tendo em conta os grupos de controlo que serão abordados. Por fim, é apresentado segundo o modelo V [4], todos os testes implementados no sistema, e a análise de usabilidade.

No sétimo capítulo é apresentado as conclusões desta dissertação, limitações e trabalho futuro.

2 Estado da arte

Este capítulo está dividido em várias secções.

Na secção 2.1 é apresentado os sistemas de classificação de informação e as respetivas abordagens.

Na secção 2.2 é apresentado os sistemas de classificação de treino desportivo.

Na secção 2.3 é apresentado o estudo e comparações feitas às soluções e abordagens semelhantes já existentes no mercado, na última subsecção é comparado os diferentes softwares e analisado relativamente ao projeto desenvolvido.

2.1 Sistemas de classificação de informação

O processo de classificação é um método de extrema importância nos sistemas baseados em conhecimento, como é o caso dos sistemas de recomendação. De seguida, são abordadas as metodologias mais relevantes de classificação de informação. Existem atualmente dois tipos de abordagens de classificação e categorização de informação, as abordagens baseadas em métodos probabilísticos (estatísticos) e abordagens baseadas em aprendizagem computacional, que serão detalhadas nas subsecções seguintes.

2.1.1 Classificação baseada em Aprendizagem

Uma das formas de realizar estes processos de classificação é utilizando técnicas de aprendizagem, ou seja, construir algoritmos que possam automaticamente e a partir de exemplos, reproduzir os mesmos efeitos de classificação a novos conteúdos ⁴.

⁴ L. Filipe; Sistema de Recomendação Baseado em Conhecimento.2012

De seguida são descritas algumas das técnicas de aprendizagem mais utilizadas neste contexto.

Support Vector Machines (Vapnik 1995) ⁵ é um método baseado em aprendizagem utilizada para classificar dados de forma supervisionada. Tecnicamente as *Support Vector Machines* (SVM) funcionam através da criação de hiper-planos num espaço multidimensional que separam objetos pertencentes a classes distintas. O objetivo central das SVM é o ajustamento, a partir de exemplos, de uma função discriminante (vetor de suporte) que maximiza a margem de separação entre as classes consideradas e a fronteira de decisão, minimizando desta forma o erro na classificação ⁶.

Naive Bayes consiste num algoritmo probabilístico pertencente à classe dos “Classificadores Bayesianos” ⁷. As *Naive Bayes* originam um modelo probabilístico baseado em observações dos dados previamente existentes, permitindo obter uma probabilidade *a posteriori* de um determinado item pertencente a uma determinada classe. Este conceito é baseado na teoria de probabilidade Bayesiana (Bayes 1763).

De uma forma simplificada podemos descrever o Teorema de Bayes na seguinte expressão representada pela Equação:

$$P\left(\frac{c}{d}\right) = \frac{P(c)P\left(\frac{d}{c}\right)}{P(d)}$$

- $P\left(\frac{c}{d}\right)$ é a probabilidade à posteriori de um item **d** pertencer à classe **c**.
- $P(c)$ é a probabilidade de observar um item na classe **c**.
- $P\left(\frac{d}{c}\right)$ é a probabilidade de observar o item **d** dada a classe **c**.
- $P(d)$ é a probabilidade de observar o item **d**.

⁵ C.Cortes, V.Vapnik; Support-Vector Networks.1995

⁶

⁷ I.Rish; An empirical study of the naive Bayes classifier.2001

Para classificar o item d , escolhe-se a classe com maior probabilidade:

$$c = \operatorname{argmax}_{c_j} \frac{P(c_j)P(d/c_j)}{P(d)}$$

$P(d)$ é normalmente removido por ser igual para todo o c_j . Como não conhecemos os valores de $P(\frac{d}{c})$ e de $P(c)$, estes são estimados a partir da observação dos dados de treino.

Decision Trees são constituídos por grafos em árvore, nos quais os nós internos estão rotulados com termos, sendo que, os ramos que deles originam estão rotulados com o peso que o termo tem no documento de teste e os nós folhas estão rotulados com categorias. As árvores de decisão são construídas dividindo sucessivamente os documentos de teste em grupos, até que estes, contenham apenas documentos pertencentes a uma determinada categoria. O teste utilizado para dividir os documentos em grupos é feito com base nos pesos que os termos contidos nos nós têm no documento ⁸.

k-Nearest Neighbour é um método de aprendizagem também conhecido por *Lazy Learning*, que aborda o problema de classificar itens novos comparando-os com a informação que está armazenada em memória (exemplos de treino) utilizando uma função de similaridade (i.e. Distância Euclidiana).

Neste processo, determinam-se os *k-nearest neighbors* (kNN), ou seja, encontram-se os *k-itens* mais próximos do novo item a classificar e é-lhe atribuída uma categoria extrapolada a partir das categorias dos seus *nearest neighbors* ⁹.

Neural Networks é constituída por uma cadeia de neurónios artificiais que imitam o comportamento dos neurónios biológicos, permitindo resolver problemas de processamento de dados com base em informação que circula dentro e fora da rede ¹⁰. Em termos claros, as redes neuronais processam os dados de entrada com base em

⁸ J. QUINLAN; Induction of Decision Trees.1986

⁹ P. Cunningham and S. J. Delany; k-Nearest Neighbour Classifiers.2007

¹⁰ J. Schmidhuber; Neural Networks.2015

modelos matemáticos que ajustam a “força” das ligações entre os neurónios para produzir um determinado resultado de saída.

RBFNetworks (Radial Basis Function Networks) são um tipo específico de redes neuronais artificiais com aplicação em problemas de aprendizagem supervisionada, nomeadamente classificação.

Este tipo de rede neuronal utiliza funções de base radial, funções não-lineares que podem ser utilizadas em modelos de regressão linear com parâmetros lineares ou não-lineares. Tipicamente as *RBF Networks* possuem três camadas, uma camada de entrada, uma camada escondida com uma função de ativação não-linear e uma camada de saída linear ¹¹.

2.1.2 Métodos estatísticos de classificação

Este tipo de abordagem pretende aproveitar os aspetos estatísticos dos elementos a classificar, mais concretamente, e no caso da classificação de documentos de texto, de aspetos semânticos dos elementos que constituem o texto. Deste modo as técnicas probabilísticas de classificação recorrem a uma série de métricas auxiliares que permitem aferir e quantificar relações entre os objetos em estudo.

De seguida, é apresentado um conjunto de métricas bastantes utilizadas para classificar objetos e/ou conjuntos de informação.

Conditional Random Fields é uma abordagem estatística utilizada para identificar ou catalogar uma sequência de dados das quais se conhecem determinadas características estruturais ¹².

Este método consiste na construção de um modelo com base nas relações estatísticas existentes entre diversas observações de um determinado padrão, permitindo assim

¹¹ M. J. L. Orr; Introduction to Radial Basis Function Networks.1996.

¹² J. Lafferty, A. McCallum and F. C. Pereira; Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.2001.

construir interpretações consistentes dessas relações que permitam identificar novos elementos que se enquadrem no modelo criado.

Inverse Document Frequency é uma medida estatística utilizada para avaliar a importância (w) de um determinado termo (t_i) para um determinado documento (d_j) num determinado universo (D) de documentos ¹³.

A métrica *Term Frequency - Inverse Document Frequency* (TF-IDF) tem como base o princípio de que a importância de um termo para um texto, de um mesmo conjunto, aumenta proporcionalmente ao número de vezes que nele ocorre, sendo diluído pelo aumento do número de vezes que o termo ocorre em outros textos do mesmo conjunto.

Podemos então descrever esta métrica do seguinte modo,

$$w(t_i, d_j) = (1 + \log_2 tf(t_i, d_j)) * \log_2 \left(\frac{|D|}{F(t_i)} \right)$$

- $D = (d_1, d_2, d_3, \dots, d_n)$ representa uma coleção contendo n elementos.
- $|D|$ número de documentos contidos numa coleção D .
- $tf(t_i, d_j)$ “*term frequency*” corresponde ao número de ocorrências de um termo num dado documento.
- $F(t_i)$ corresponde ao número de documentos onde t_i ocorre.

Co-citation é uma métrica de correlação utilizada para medir a semelhança entre documentos ¹⁴. É um método que procura identificar semelhanças nos documentos científicos através da análise das referências.

¹³ K. Papineni; Why Inverse Document Frequency?.2001

¹⁴ M. J. Culnan; The Intellectual Development of Management Information Systems, 1972–1982: A Co-Citation Analysis. 1986

Term Cooccurrence (coocorrência) corresponde genericamente à ocorrência de dois termos num mesmo corpus de texto, tendo como base a *Distributional Hypothesis* que propõe que palavras que ocorrem num mesmo contexto revelam tendência para possuir o mesmo significado ¹⁵.

2.2 Sistemas de classificação de treino desportivo

A classificação dos exercícios de treino faz-se com base em dois critérios fundamentais. O conteúdo de treino predominante e o grau de identidade do exercício com a realidade da competição ¹⁶.

Tabela 1 Classificação dos exercícios de treino ¹⁷.

Exercício de Treino	Técnicos	Exercícios para aprendizagem, aperfeiçoamento e melhoria dos aspetos técnicos da modalidade desportiva.
	Táticos	Orientados para a construção do repertório de estratégias de cariz tático. Modelos de organização defensiva, ofensiva e de transição no caso dos jogos desportivos coletivos e de gestão do esforço em modalidades individuais.
	Físicos	Destinam-se a melhorar os fatores físicos determinantes do desporto.

No treino contemporâneo quase não existem situações em que o treino seja só técnico, tático ou físico. O mais usual é haver uma seleção de tarefas de treino técnico e tático com a exigência de um regime de exigência física específico e controlado. Assim faz-se a estimulação da execução técnica com um enquadramento tático em regimes energéticos associados. Um exemplo que se pode dar, é o treino técnico-tático em regime de preparação física (condução e remate em regime de velocidade) ou o treino

¹⁵ I. Dagan, L. Lee and F. C. N. Pereira; Similarity-Based Models of Word Cooccurrence Probabilities. 1999

¹⁶ L. Rama; Teoria e Metodologia do Treino. 2016

¹⁷ L. Rama org.

de preparação física em regime de execução técnica-tática (resistência em regime de marcação individual).

Exercícios Gerais

Os exercícios gerais têm por objetivo a formação e a multilateralidade no intuito do desenvolvimento global e harmonioso dos atletas, compensando as desvantagens de excessiva especificidade na estimulação do treino.

De forma resumida, os exercícios gerais pretendem atingir vários objetivos como uma maior tolerância à carga de treino; trazem aspectos motivacionais para o processo de treino e contribuem para a construção de um património individual em termos técnicos e táticos.

Exercícios Específicos

Estes exercícios podem ser divididos, em preliminares ou de aproximação e exercícios de desenvolvimento que têm por objetivo fundamental o desenvolvimento do fator físico. Muitos desses exercícios são realizados em situação de competição.

Exercícios Competitivos

Os exercícios competitivos correspondem à realização em situação de competição real ou simulada. Pretendem garantir o conjunto das adaptações necessárias para a melhoria do desempenho em situação de competição.

Desempenham um papel determinante, uma vez que permitem usufruir de estimulações idênticas à da realidade e os requisitos da situação de competição.

Tabela 2 Classificação dos exercícios de treino em função da identidade ¹⁸.

Exercício de treino	Gerais	Distinguem-se da execução comum em competição. Tem um carácter complementar no processo de treino, quer preenchendo lacunas, quer favorecendo a recuperação nos domínios físico e mental.
	Especiais	Correspondem nos recursos energéticos. Utilizam exigências de carga de treino semelhantes às da competição.
	Competitivos	Competição formal ou competição simulada – provas de preparação ou jogos amigáveis.

2.3 Softwares existentes de apoio ao treinador

Nesta secção serão identificados cinco softwares existentes no mercado, que de certo modo, auxiliam treinadores de futebol a gerir a sua equipa. É importante salientar que não foram encontrados softwares destinados unicamente a atletas que permitissem de forma direta melhorar as suas performances, no entanto, algumas das aplicações estudadas, são também dirigidas aos jogadores, possibilitando assim, a visualização e análise dos exercícios de treino propostos.

2.3.1 Metodologia de pesquisa

Os softwares analisados foram obtidos através da procura nos motores de pesquisa, (*Google e Apple Store*) dos softwares mais usados nesta área em Portugal, em que os

¹⁸ L. Rama org.

mesmos tivessem uma classificação de avaliação superior a quatro (4), na escala de um a cinco (1-5). Os softwares obtidos, dividem-se entre aplicações móveis e aplicações browser/desktop e aplicações móveis que estão disponíveis tanto para sistemas operativos IOS, como Android. Foram apenas apresentadas cinco aplicações, uma vez que, por norma, os softwares encontrados não apresentam funções exclusivas, ou seja, os softwares são bastantes semelhantes entre si, oferecendo aos seus utilizadores funcionalidades base idênticas.

De seguida, para cada um dos softwares analisados, serão primeiramente detalhadas as suas funcionalidades, apresentando as suas principais qualidades. Após ser descrita cada aplicação, será realizada uma comparação dos cinco softwares expostos, através de uma tabela, concluindo qual dos mesmos será o mais eficiente para o treinador/jogador (caso destinada a esta entidade) efetuar o seu trabalho.

Esta análise foi realizada com o objetivo de retirar ideias sobre possíveis funcionalidades a acrescentar ao sistema, assim como, constatar que a adição do sistema de classificação de exercícios por atributo será uma funcionalidade singular e única na aplicação a desenvolver.

2.3.2 TacticalBorders soccer

É uma aplicação nativa que pode ser utilizada em qualquer dispositivo seja ele móvel ou não. Foi desenvolvida com o objetivo de melhorar o treino e reduzir o tempo necessário na preparação do mesmo. Este software foi desenvolvido com a ajuda de treinadores, jogadores, pais, academias e clubes, de forma a incluir todas as ferramentas necessárias para preencher todas as necessidades dos formadores e dos jogadores.

Tem como principais funcionalidades, criar exercícios, preparar bolas paradas, sessões de treino, opções táticas, definir os convocados e a estratégia, e muito mais. Contém uma biblioteca de exercícios que está segura e acessível a qualquer momento, a mesma com dados referentes a exercícios / treinos / estratégias pré jogo / análises pós-jogo.

Na funcionalidade de criar exercício existem ferramentas de animação de exercícios para auxiliar, permitindo assim ao utilizador ter uma ideia como irá funcionar o exercício no campo. Todos os exercícios criados podem ser gravados numa base local, e assim reutilizados noutros planos de treinos.

Ideal para equipas que partilham filosofias e metodologias de treino por vários treinadores, uma vez que permite a partilha de exercícios técnicos e táticos entre vários utilizadores.



Figura 2 Tactical Borders Soccer ¹⁹

2.3.3 Coach-helper

Coach-Helper é um software exclusivo para computadores, que foi desenvolvido para auxiliar os treinadores na preparação do treino. O software contém uma base de dados de exercícios que já estão definidos, que podem ser utilizados pelos utilizadores para criar o seu plano de treinos, fornece também a possibilidade de editar os exercícios e guardar sem alterar os originais. Existe a possibilidade de criar uma base de dados de exercícios pessoal e privada.

A funcionalidade base do software é criar planos de treinos específicos para o treinador, com base nos exercícios selecionados pelo utilizador, ou exercícios criados pelo próprio.

¹⁹ Site: <https://www.tacticalsoccer.pt>

Como referido, o utilizador pode criar exercícios através do editor presente no software, sendo que existe a possibilidade de fazer *upload* de imagens/vídeos para exemplificar os exercícios. O software cria planos de treino com/sem croquis, existindo a possibilidade de o utilizador guardar o plano de treino nos favoritos para reutilizar no futuro.

O software permite toda a gestão do treino, mas também dos jogadores. O utilizador pode criar fichas individuais dos jogadores com todos os dados necessários.

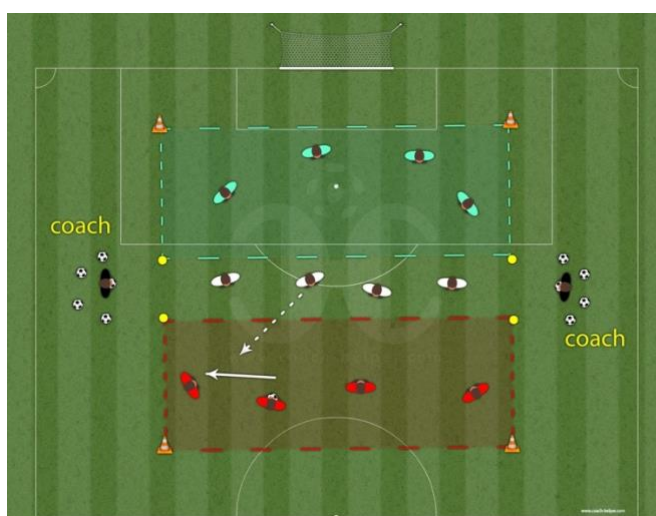


Figura 3 Coach-Helper ²⁰

2.3.4 MyCoach

MyCoach é uma plataforma híbrida, isto é, funciona nos dispositivos móveis e no computador, o recurso de treino é uma autêntica biblioteca especializada. O utilizador pode criar e partilhar as folhas de exercícios, tem a possibilidade de organizar uma única sessão ou planear uma série de treinos num determinado período. O utilizador poderá adicionar conteúdo de acordo com os temas que deseja abordar.

²⁰ Site: <https://www.coach-helper.com>

No My Coach Football, existem mais de quatro mil (4000) cartões de exercícios, classificados por tema, categoria e idioma. Todos foram criados por treinadores desde a criação da plataforma.

O utilizador pode criar o próprio plano de treinos. Para isso, seleccione o nome da atividade, a categoria de idade a que se destina, mas também o idioma utilizado, os temas principais e secundários, os objetivos e, finalmente, o número de jogadores necessários. O utilizador terá a oportunidade de inserir o material que será usado, as dimensões do campo necessário, as instruções para aplicar e os critérios de realização para respeitar. Define também a duração do esforço, a intensidade, o número de repetições e séries necessárias e o tempo e a natureza da recuperação.

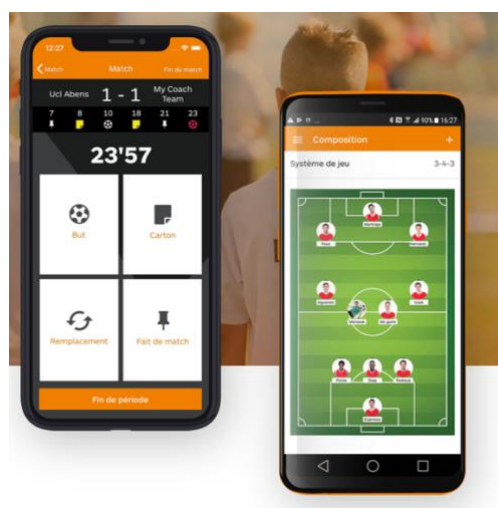


Figura 4 MyCoach Football ²¹

2.3.5 Soccer-trainer

Soccer-trainer é uma plataforma web, que é suportado em todos os navegadores web. O software permite ao utilizador escolher sessões de treinos disponíveis na base de dados, mas também tem a possibilidade de modificar e adaptar os programas. As sessões estão classificadas por grupo disciplinar e idade. O software tem a funcionalidade que permite ao utilizador criar os próprios exercícios, definindo o

²¹ Site: <https://www.mycoachfootball.com>

objetivo, os materiais necessários, a descrição e uma imagem ilustrativa do mesmo. A base de dados contém mais de trezentos e cinquenta (350) exercícios, ordenados por atributo.

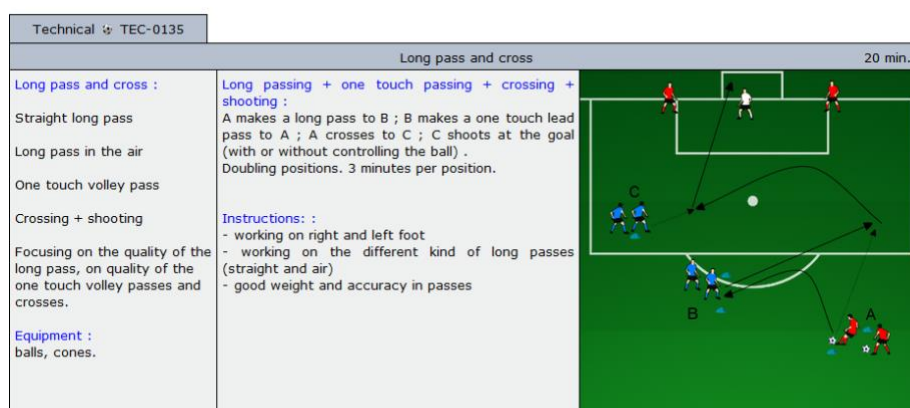


Figura 5 Soccer-Trainer Online ²²

2.3.6 Newsoccerdrills

Newsoccerdrills é um software disponível para o computador, smartphone/tablet. Contém uma enorme quantidade de planos de treinos profissionais que podem ser personalizados, exercícios com gráficos e vídeos dos treinos dos principais clubes europeus. O software tem um sistema de pesquisa integrado com a possibilidade de selecionar os exercícios por objetivos, idade/número de jogadores, espaço, etc....

Newsoccerdrills distingue-se dos demais, por possuir a opção de geração de um plano de treino, para isso o utilizador só tem de selecionar os objetivos específicos, e o software vai apresentar um plano de treino profissional com base nas informações fornecidas.

²² Site: <http://www.soccer-trainer.com.pt>



Figura 6 New Soccer Drills ²³

2.3.7 Comparação entre funcionalidades

Com base nos softwares analisados anteriormente, vão ser analisadas as principais funcionalidades, com o objetivo de averiguar se existem softwares com o mesmo propósito do implementado para o projeto. E assim, se possível concluir se o software é único no mercado atual.

A tabela seguinte contém um resumo dos requisitos e funcionalidades dos exemplos acima mencionados.

Tabela 3 Tabela Comparativa das funcionalidades

Softwares	Perfis Individuais	Base de dados de exercícios	App Mobile	Exercícios com base em fatores (idade)	Geração automática do plano de treinos
TacticalBorders soccer	X	X	X		
Coach-helper	X	X	X		
Mycoach	X	X	X		
Soccer-trainer		X			
Newsoccerdrills		X		X	X

²³ Site: <https://www.newsoccerdrills.com>

Analisando a tabela em cima podemos verificar que existem três (3) softwares dos analisados que são muito idênticos em termos de funcionalidades, caracterizando-se por serem softwares mais focados no desenvolvimento de exercícios de forma interativa e apelativa, gestão de equipas e jogador, possibilitando ao utilizador criar o seu próprio plano de treino com os exercícios criados por eles. O software que vai mais de encontro com o projeto é o Newsoccerdrills, este, caracteriza-se por conter uma base de dados com exercícios já definidos por treinadores, sendo que, o que mais valoriza este software é conter exercícios com base em fatores, neste caso, o fator é a idade. A funcionalidade base deste software é a geração automática de planos de treino para o utilizador, permitindo depois a personalização do plano gerado.

3 Análise de Valor

A análise de valor é um processo sistemático, formal e organizado de análise e avaliação. Não é casual ou informal e é uma atividade de gestão que requer planeamento, controlo e coordenação.

O principal objetivo de uma análise de valor passa por maximizar o valor de um produto ou serviço, ao menor custo sem sacrificar a qualidade. Caracteriza-se por ser uma filosofia implementada pelo uso de um conjunto específico de técnicas, uma base de conhecimento, e um grupo de habilidades. É uma abordagem organizada que tem como finalidade a identificação eficiente dos custos desnecessários, ou seja, o custo que não oferece nem qualidade, nem uso, nem aparência, nem características do cliente.

Como tal, será explicado inicialmente nesta secção, o processo de inovação de Peter Koen que se denomina Novo Modelo de Desenvolvimento de Conceito (NCD) ²⁴, aplicá-lo ao projeto. De seguida, vão ser explicados os conceitos bases de Valor, Valor Percecionado e Valor para o cliente e identificado através de uma representação com o auxílio de uma tabela os benefícios e os sacrifícios para o cliente. Por fim, é apresentado a proposta de valor do projeto para o cliente e o modelo de negócios assente no modelo CANVAS.

²⁴ PeterA.Koen, Fuzzy Front End: Effective Methods, Tools, and Techniques. 2002

3.1.1 Novo Modelo de Desenvolvimento de Conceito (NCD)

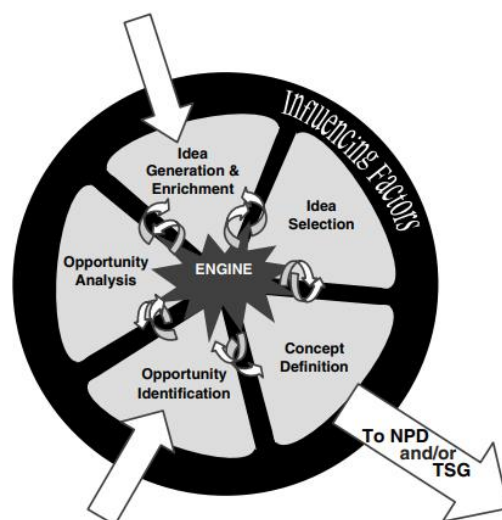


Figura 7 O modelo New Concept Development (NCD)²⁵

O modelo é constituído por três partes chave:

1. O motor, representa a liderança, cultura e estratégia de negócio da organização, o mesmo, serve ainda de apoio aos elementos existentes na área interna;
2. As áreas internas, representam os cinco elementos chave. Nesta área, as ideias e conceitos devem iterar entre os cinco elementos;
3. Os fatores externos, que são usualmente incontroláveis pela organização, mas podem influenciar o processo de inovação da mesma, nomeadamente na fase de comercialização;

Na figura exposta acima, é ainda possível concluir que as setas de entrada representam os possíveis pontos de partida no desenvolvimento de um projeto, nomeadamente, identificação de oportunidades e/ou geração de ideias. Já em relação, à seta de saída, esta representa um conceito de desenvolvimento de produto a sair do modelo, quer pelo processo de New Product Development (NPD), ou pelo processo Technology State Gate (TSG).

²⁵ Peter A. Koen, Fuzzy Front End: Effective Methods, Tools, and Techniques. 2002

Desse modo, podemos concluir que o modelo NCD contém cinco elementos chave presente na área interna, entre eles: identificação de oportunidades, análise de oportunidades, geração de ideias, seleção de ideias e definição de conceito. De seguida, são esclarecidos cada um destes elementos e como são aplicados ao presente projeto.

3.1.1.1 Identificação de Oportunidade

Neste conceito, a organização identifica as oportunidades que deseja seguir. As oportunidades comerciais e tecnológicas são explicitamente consideradas, de modo a que, os recursos sejam alocados para novas áreas de crescimento de mercado, eficácia operacional e eficiência. Esse elemento é geralmente impulsionado pelos objetivos do negócio. Por exemplo, a oportunidade pode ser uma resposta de curto prazo a uma ameaça competitiva, uma possibilidade “inovadora” de capturar vantagem competitiva ou um meio de simplificar operações. Pode ser uma direção totalmente nova para o negócio ou uma atualização para um produto já existente.

Aplicando este conceito ao projeto, identificou-se a oportunidade de criar um software com a capacidade de classificar os exercícios de treino através dos atributos, filtrar os mesmos e assim posteriormente aplicar o conceito de um sistema de recomendação. Esta identificação teve em atenção ao crescente aumento da importância das novas tecnologias no desporto em geral, mas mais especificamente no futebol, para conseguir obter vantagens em relação aos rivais. Outro ponto importante na identificação, foi o aumento substancial de aplicações no desporto, sendo que a maioria delas relativas a estatísticas e registo/visualização de dados.

3.1.1.2 Análise da Oportunidade

Neste conceito, uma oportunidade é avaliada para garantir que vale a pena prosseguir. Informações adicionais são necessárias para traduzir a identificação de oportunidades em oportunidades específicas de negócio e tecnologia. Isso implica fazer avaliações precoces e muitas vezes incertas sobre tecnologia e mercado. O esforço despendido

dependerá do valor das informações associadas à redução das incertezas sobre a atratividade da oportunidade, o tamanho esperado do esforço de desenvolvimento futuro, dada a adequação com a estratégia e cultura de negócios e o risco dos responsáveis pelas tomadas de decisão. A análise de oportunidades pode fazer parte de um processo formal ou pode ocorrer de forma iterativa. A capacidade e a competência do negócio são avaliadas neste elemento.

Aplicando este conceito ao projeto, é reconhecido através de uma análise ao mercado que, não existem aplicações com um sistema de classificação de exercícios por atributos, que permita a integração com um sistema de recomendação de planos de treino, o que classifica este projeto como único.

3.1.1.3 Geração e melhoramento de ideias

O conceito de geração e melhoramento de ideias está associado ao nascimento, desenvolvimento e maturação de uma ideia concreta. A geração de ideias é evolutiva. As ideias são construídas, desmontadas, combinadas, reformuladas, modificadas e atualizadas. Uma ideia pode passar por muitas iterações e mudanças à medida que é examinada, estudada, discutida e desenvolvida em conjunto com outros elementos do modelo de DCNT.

A geração e o melhoramento de ideias podem ser um processo formal, incluindo sessões de *brain-storming*, de modo a impulsionar a organização na geração de novas ideias ou modificá-las para a oportunidade identificada. Uma nova ideia também pode surgir fora dos limites de qualquer processo formal - como uma experiência que dá errado, um fornecedor que oferece um novo material ou um utilizador que fez uma solicitação incomum. A geração e o melhoramento de ideias podem alimentar a identificação de oportunidades, demonstrando assim que os elementos das DNTs geralmente avançam de forma não-linear, e alimentam ideias onde quer que elas ocorram.

Aplicando este conceito ao projeto, foi identificado um conjunto de ideias, tais como, desenvolver uma aplicação web e mobile, desenvolver um sistema de gestão para equipas de futebol, implementar um mecanismo de controlo de dados individuais e coletivos, desenvolver um sistema de classificação de exercícios por atributos, e por último um sistema de gestão de exercícios.

3.1.1.4 Seleção de ideia

Na maioria dos casos, o problema não está no surgimento de novas ideias. Mesmo quando as empresas estão a ser reduzidas, não faltam novas ideias. O problema para a maioria das empresas é selecionar as ideias a seguir para alcançar o maior valor de negócios. Fazer uma boa seleção é fundamental para a saúde e o para o sucesso futuro do negócio. No entanto, não existe um processo único que garanta uma boa seleção. A maioria das seleções de ideias envolve uma série iterativa de atividades que provavelmente incluirá várias etapas através da identificação de oportunidades, análise de oportunidades e geração e melhoramento de ideias, geralmente com novos fatores de influência e novas direções do mecanismo ²⁶.

Aplicando este conceito ao projeto, as ideias que foram consideradas mais importantes, ou seja, que têm mais valor para os interessados (treinadores, jogadores, staff), são as seguintes: Aplicação web responsiva, sistema de gestão para equipas de futebol, sistema de gestão de exercícios, sistema de classificação de exercícios por atributos e mecanismo de controlo de dados coletivos e individuais.

3.1.1.5 Definição de conceito

Definição de conceito é o elemento final do novo modelo de desenvolvimento de conceito. Esse elemento fornece a única saída para o NPD ou o TSG (technology stage gate). Para passar pelo portão, o inovador deve apresentar argumentos convincentes para o investimento no negócio ou na proposta de tecnologia. Algumas organizações referem isso como uma “declaração de ganho”, outros chamam de “documento de

²⁶ Peter A. Koen, *Fuzzy Front End: Effective Methods, Tools, and Techniques*. 2002

porta”. O caso de investimento consiste em informações qualitativas e quantitativas, que os guardas usam para fazer uma determinação. A maioria das empresas especifica diretrizes para os guardas, que tomam decisões no início do processo de desenvolvimento. Estes podem abordar:

- Objetivos
- Encaixe do conceito com estratégias corporativas e / ou divisionais
- Tamanho da oportunidade, como impacto financeiro
- Mercado ou necessidades e benefícios do cliente
- Um plano de negócios que especifica uma proposta de valor específica de “*win/win*” para participantes da cadeia de valor
- Fatores de risco comerciais e técnicos
- “*showstoppers*” ambientais, de saúde e segurança
- Um plano de projeto que inclui recursos e tempo

Os requisitos e critérios de informação variam de acordo com a natureza e com o tipo de conceito, bem como das atitudes dos responsáveis pela tomada de decisão em relação ao risco.

Aplicando este conceito ao projeto, como não existem muitos riscos associados a este, o principal objetivo foi criar um plano de trabalho exequível no tempo disponível com os recursos existentes.

3.1.2 Valor, Valor Percecionado e Valor para o cliente

O processo de desenvolvimento de uma nova aplicação tem de apresentar como um dos principais objetivos a criação de novos valores, para garantir assim o sucesso do negócio. De modo a analisar esses valores, serão apresentados nesta subsecção três

conceitos relevantes nesta área, nomeadamente, valor, valor percebido e valor para o cliente. Por fim, serão identificados os principais benefícios e sacrifícios para o cliente, de acordo com o presente projeto.

3.1.2.1 Valor

O valor é uma quantidade que aumenta a satisfação do cliente ou reduz o gasto inerente ao produto, como tal é um parâmetro essencial para a melhoria do processo. A criação de valor é fundamental para o sucesso de qualquer negócio, seja qual for a atividade de negócio, visto que é a ligação entre a satisfação do cliente e o preço. Qualquer atividade de negócio prende-se com a troca de valor com os seus clientes, seja ela através de um bem ou através de um serviço, quer seja tangível e/ou intangível.

3.1.2.2 Valor Percebido

O valor percebido pelo produtor tem um significado de valor diferente ao percebido pelo cliente, ou seja: o produtor é menos sensível ao preço, enquanto o consumidor é mais sensível à qualidade do produto.

3.1.2.3 Valor para o cliente

O valor para o cliente é uma avaliação realizada pelos clientes relativamente à relação existencial entre os benefícios e os sacrifícios de um determinado valor. Posto isto, é de realçar que os clientes usufruem de produtos ou serviços não só para a sua satisfação, mas também para a sua própria necessidade.

3.1.2.4 Benefícios e Sacrifícios para o cliente

O benefício deste projeto para o cliente é o desenvolvimento de uma aplicação intuitiva para auxiliar treinadores na sua atividade de gestão da sua equipa e jogadores, e melhorar a suas performances através de um mecanismo de classificação de exercícios por atributos, que vai ser integrado com um sistema de recomendação de planos de

treino, para além disso deve estar presente um mecanismo de gestão de exercícios de treino.

Relativamente aos sacrifícios para o cliente, poder-se-á considerar o custo das licenças do software.

Por fim, de modo a sumarizar a avaliação da relação entre os benefícios e os sacrifícios, aplicados ao presente projeto, serão apresentados na Tabela 4 o resumo dos mesmos.

Tabela 4 Benefícios e Sacrifícios para o cliente

Benefícios	Sacrifícios
Aplicação inovadora	Custo
Interfaces completas e intuitivas	-
Funcionalidades de gestão	-
Classificações de exercícios	-
Gestão de exercícios	-

3.1.3 Proposta de Valor

A proposta de valor é entendida como o conjunto de produtos e serviços que criam valor para um segmento específico de cliente (Alex Osterwalder). A proposta de valor deve descrever o produto da maneira mais simplificada e clara possível, diferenciando o mesmo dos seus concorrentes diretos, de maneira a expor o motivo pelo qual o cliente deve escolher o produto ou marca concreta da empresa, e não o da concorrência.

A proposta de valor para o presente projeto:

Oferta de uma aplicação destinada a treinadores de futebol que disponibiliza suporte técnico para gerir toda sua equipa ao longo de uma época desportiva, a mesma é capaz de melhorar a performance dos jovens jogadores através da classificação de exercícios e planos de treinos, que posteriormente através de um sistema de recomendação com o uso do sistema de classificação irá gerar planos de treinos específicos para as necessidades dos respetivos atletas.

3.1.4 Modelo Canvas

Um modelo de negócio descreve como uma empresa cria, entrega e captura valor. As empresas utilizam estes modelos de maneira a ter uma visão estruturada de todos os componentes que constituem o negócio.

O modelo de negócio utilizado para este projeto foi o Modelo Canvas²⁷, que permite uma visão holística do mesmo. É constituído por 9 componentes fundamentais para definir o funcionamento do negócio:

- A proposta de valor do que é oferecido ao mercado (***Value Proposition***);
- Os segmentos de clientes a quem se destina a proposta de valor (***Customer Segments***);
- Os canais de distribuição e comunicação usados para transmitir aos clientes a proposta de valor e o respetivo produto (***Channels***);
- As relações estabelecidas com os clientes (***Customer Relationships***);
- As atividades chave necessárias para implementar o negócio (***Key Activities***);
- Os parceiros chaves que participam no negócio (***Key Partners***);

²⁷ M.Zilber, F.Bonazzi; Innovation and Business Model: a case study about Integration of Innovation Funnel and Business Model Canvas. 2014

- Os recursos chaves necessários para concretizar o negócio (**Key Resources**);
- A estrutura dos custos resultantes da criação e manutenção do negócio (**Cost Structure**);
- As fontes de receita geradas pelo negócio (**Revenue Streams**);

Com base nestas considerações e utilizando a estrutura definida pelo Modelo Canvas, foi criado a seguinte figura:

Key Partners <ul style="list-style-type: none"> • ISEP • Clubes Desportivos 	Key Activities <ul style="list-style-type: none"> • Design do Projeto • Desenvolvimento do projeto • Testes ao projeto • Suporte Online • Manutenção 	Value Propositions <ul style="list-style-type: none"> • Gestão de equipas de futebol • Gestão dos planos de treino • Criação personalizada de exercícios • Filtragem dos exercícios por atributo • Exercícios classificados por atributos 	Customer Relationships <ul style="list-style-type: none"> • E-mails • Redes Sociais • Notificações Móveis 	Customer Segments <ul style="list-style-type: none"> • Jogadores de Futebol • Treinadores de Futebol
	Key Resources <ul style="list-style-type: none"> • Equipamento Informático • Licenças de softwares para o desenvolvimento do produto • Servidores 		Channels <ul style="list-style-type: none"> • Clubes desportivos • Site oficial 	
Cost Structure <ul style="list-style-type: none"> • Custos da plataforma • Custos da manutenção • Custos da Publicidade 			Revenue Streams <ul style="list-style-type: none"> • Vendas de licença 	

Figura 8 Modelo Canvas

4 Análise e Design

Neste capítulo encontra-se uma explicação das decisões tomadas na conceção do software, tem como objetivo descrever o design realizado para a solução proposta. Desse modo, primeiramente, será detalhada a visão da solução que contempla uma proposta de resolução para o problema descrito.

De seguida, é exposta toda a análise e conceção da solução, esta, contém a identificação dos *stakeholders*, atores do sistema e a descrição dos requisitos funcionais e não funcionais.

Posteriormente, é apresentada a arquitetura da solução, através do modelo de vistas quatro mais um (4+1), identificando os seguintes componentes: vista lógica, vista de implementação, vista de implantação, vista de dados e vista dos casos de uso.

Por fim, são ainda identificadas possíveis futuras maquetes da interface da aplicação.

4.1 Visão da Solução

Nesta secção, realiza-se uma análise á visão da solução elaborada de maneira a dar resposta ao problema identificado na secção **1.2 – Problema**.

De maneira a solucionar o problema, é proposto a criação de uma aplicação web responsiva que vai ser integrada com o sistema de classificação de exercícios por atributo. No desenvolvimento do sistema de classificação de exercícios por atributos é utilizada a abordagem baseada em aprendizagem computacional, ou seja, é proposto que seja desenvolvido um algoritmo que possa automaticamente e a partir de exemplos, reproduzir os mesmos efeitos de classificação de novos exercícios. A classificação dos exercícios tem em conta os critérios definidos e explicados na secção **2.2 - Sistemas de classificação de treino desportivo**. O sistema de classificação vai armazenar os dados numa base de dados própria, para posteriormente ser utilizada por um sistema de recomendação de planos de treino.

Para uma melhor compreensão da solução elaborada, é exposto todo o processo de classificação da informação pelo sistema, através de um diagrama de processo.

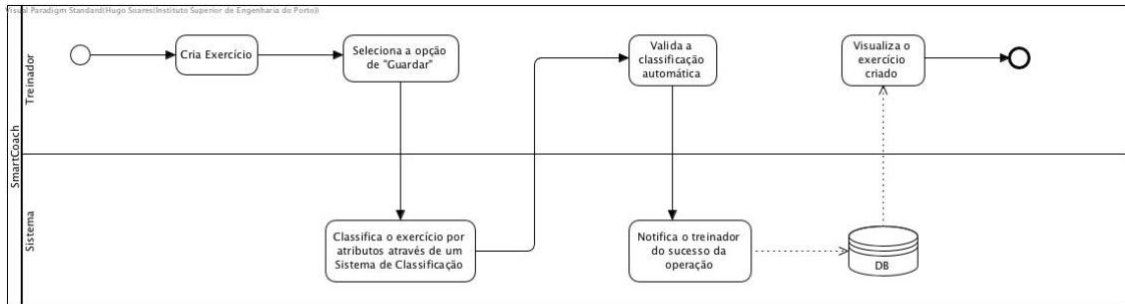


Figura 9 Diagrama de processo de Classificação

Analisando o processo de classificação representado pelo diagrama anterior, verifica-se que o utilizador da aplicação inicia a criação de um exercício de treino e insere todos os dados necessários para o mesmo, de seguida seleciona a opção de armazenar na base de dados o exercício. Em backend, o sistema recebe a informação enviada pelo utilizador e com base num algoritmo implementado de classificação de informação, o sistema classifica o exercício com base num atributo. Por fim, o utilizador valida a classificação automática, e o sistema notifica o mesmo do sucesso da operação. O utilizador, tem a possibilidade após esta operação, de visualizar o exercício criado e a respetiva classificação.

4.2 Análise e Conceção

Nesta secção vai ser analisado os Stakeholders e os Atores referentes a este projeto. De seguida, vão ser abordados os requisitos funcionais, auxiliados com o diagrama de casos de uso para cada ator identificado e a respetiva descrição, sendo também abordados os requisitos não funcionais. Por fim, será apresentado o modelo de domínio referente ao projeto.

4.2.1 Stakeholders

Stakeholders ou partes interessadas, são consideradas uma entidade que poderá ser afetada, afetar ou simplesmente ter interesse, de forma direta ou indireta, positiva ou negativamente pelo sistema. Assim sendo, é importante identificar quem são essas entidades, uma vez que estas poderão influenciar o processo de desenvolvimento do sistema.

Desse modo, foram identificados os seguintes stakeholders:

- Equipas de Futebol: Entidade que pretende utilizar a aplicação desenvolvida internamente.
- Orientador: Pessoa encarregue de orientar o estudante na construção e no desenvolvimento do projeto.
- Estudante: Entidade responsável pelo desenvolvimento da aplicação.
- Treinador: Pretende obter bons resultados desportivamente e gerir o bom funcionamento da aplicação.
- Staff Técnico: Pretende ajudar o treinador a gerir a sua equipa.
- Jogador: Pretende melhorar as suas performances desportivas.
- Utilizador da aplicação: Pretende aceder a aplicação de forma simples e intuitiva, com vista a ajudar o jogador a melhorar as suas qualidades.

4.2.2 Atores

Um Ator do sistema é considerado um utilizador que desempenha um papel específico na aplicação. Assim sendo, foram identificados os seguintes Atores do sistema:

- **Administrador:** Utilizador com acesso a todas as funcionalidades do *software*, sendo o único utilizador com permissão para fazer CRUD dos atributos e

exercícios “bases”, que são partilhados com todos os utilizadores registados na aplicação.

- **Staff Técnico/Treinador:** Utilizador que tem permissão para aceder às funcionalidades do software, tais como, gestão de atributos, exercícios e treinos, visualizar o calendário de treinos e aceder a informações fornecidas na forma de tabelas estatísticas.
- **Utilizador não registado:** Utilizador que não se encontra registado no sistema, que possui a funcionalidade básica de efetuar registo na aplicação.

4.2.3 Modelo de Domínio

O modelo de domínio é um artefacto que permite descrever e modelar entidades do mundo real e as relações entre elas, que, coletivamente, descrevem o espaço de domínio do problema. Os conceitos representados num modelo de domínio, são classes candidatas que podem, ou não, transitar para um modelo de classes. ²⁸

Para enquadrar o leitor nos conceitos apresentados no modelo de domínio da Figura 10, apresentam-se em seguida as suas definições, que são:

- **Staff técnico:** representa qualquer elemento que pertença ao staff técnico de uma equipa.
- **Treinador:** representa um Treinador. Está associado a uma equipa e a um staff técnico, e é responsável pela gestão da equipa e tudo o que engloba o treino.
- **Equipa:** representa uma Equipa de futebol. É constituída por treinadores, staff técnico e jogadores.
- **Treino:** representa um treino de uma equipa de futebol ou jogador. Este treino é composto por vários **Exercícios** e pode estar agregado a uma **Recomendação**.

- **Exercício Treino:** representa um exercício de treino que irá ser realizado por uma equipa ou jogador. Este exercício é composto por vários **AtributoClassificação**. Ao longo da realização do exercício é também notório o melhoramento destes mesmos atributos. E pode conter um ou mais **Ficheiros**.
- **Atributo Classificação:** representa a classificação do atributo quanto ao exercício associado, a classificação corresponde a um intervalo de 0 a 5. Esta classe contém um **Atributo**.
- **Atributos:** representa uma capacidade ou característica de um jogador de futebol.
- **FicheiroExercicio:** representa um ficheiro que pode ser uma imagem, vídeo, ou outro tipo de ficheiro.
- **Recomendação:** representa uma recomendação de atributos a melhorar consoante uma análise às estatísticas de um jogador. É composto por um ou vários **AtributoRecomendação** e está associado a um **JogadorRecomendação**.
- **Jogador Recomendação:** representa o jogador a ser analisado para obter uma recomendação.
- **Atributo Recomendação:** representa o atributo que vai ser recomendado melhorar e o respetivo valor de recomendação.
- **Plano de treino:** representa um plano de treino de uma equipa ou jogador. É composto por vários **Treinos**.

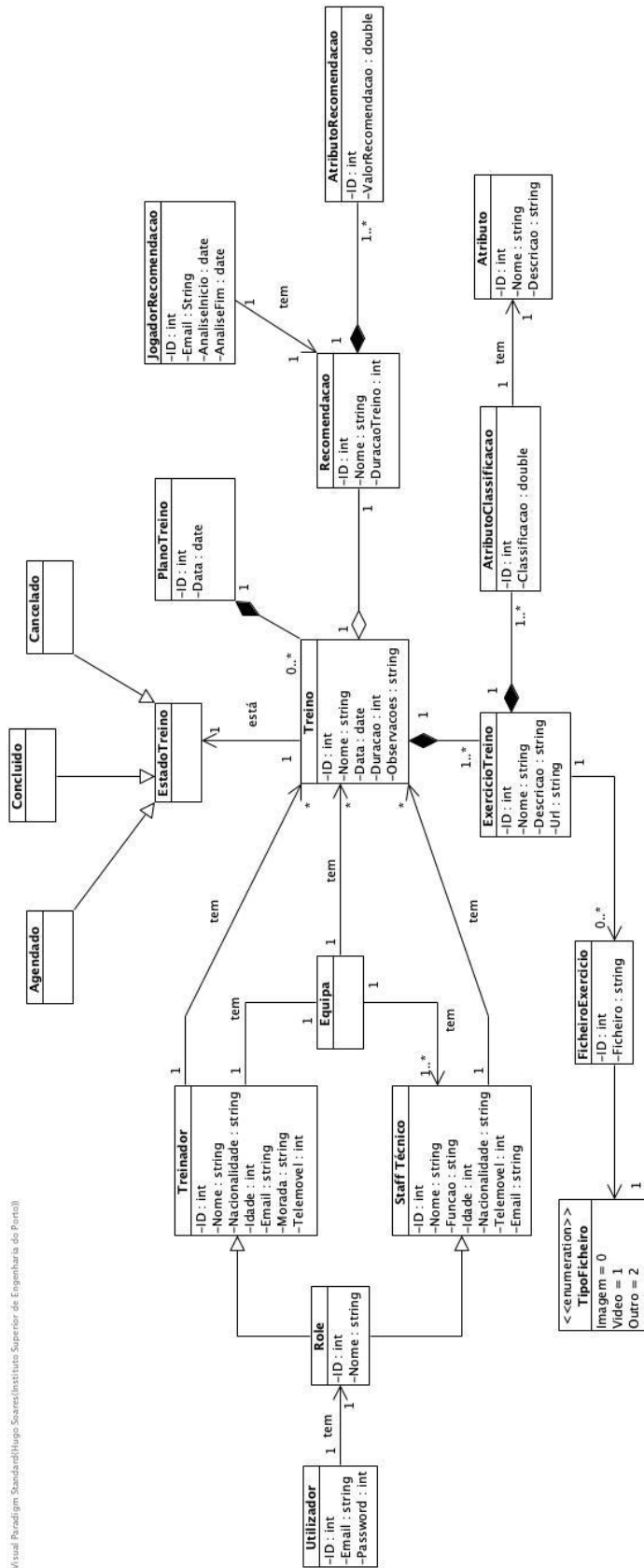


Figura 10 Modelo de Domínio

4.2.4 Requisitos funcionais

Nos requisitos funcionais, serão enumeradas as funcionalidades identificadas ao longo do projeto que, por sua vez, deram origem à criação de casos de uso. Primeiramente será apresentado o diagrama de casos de uso diferenciando os casos de uso pelos respectivos atores, e de seguida, será exposta uma breve descrição dos mesmos.

4.2.4.1 Diagrama de Casos de Uso

Com base nos requisitos levantados e tendo em conta os atores identificados na **secção 4.2.2**, foram identificados os seguintes casos de uso:

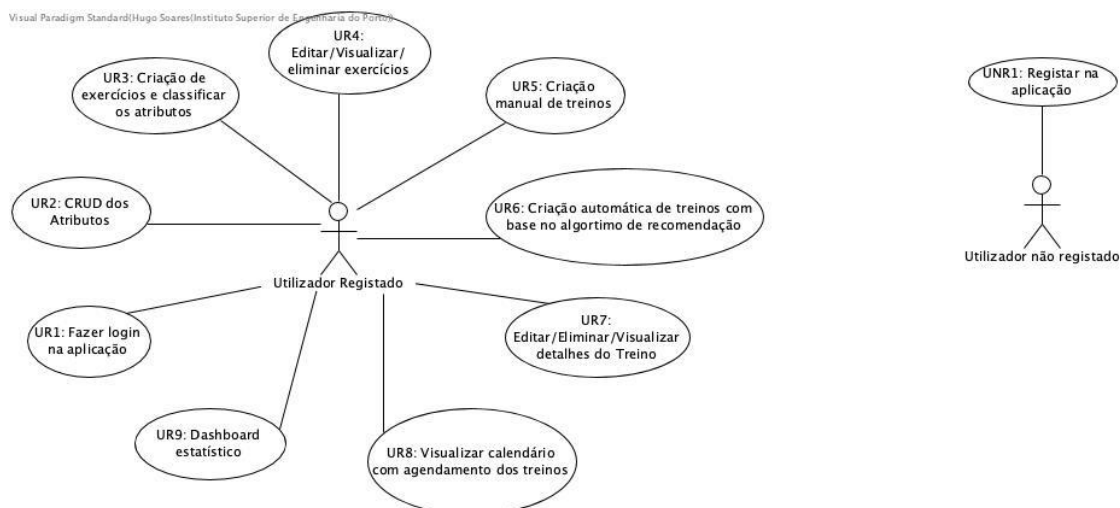


Figura 11 Diagrama de casos de uso para os atores identificados

4.2.4.2 Casos de uso do Utilizador Registado

UR01 – Fazer login na aplicação: O utilizador registado inicia o processo de *login* na aplicação, introduz os dados necessários para o mesmo e finaliza o processo.

UR02 – CRUD dos Atributos: O utilizador registado seleciona a opção Atributos no menu da aplicação, onde terá um novo painel aonde deverá indicar a ação que pretende realizar, as opções disponíveis são: visualizar detalhes de atributo(s), criar novo(s) atributo(s), editar dados do atributo(s) ou remover atributo(s).

UR03 – Criação de exercícios e classificar os atributos: O utilizador registado seleciona a opção de “Criar novo exercício”, introduz os dados solicitados pelo sistema, tais como, nome, descrição, duração em minutos do treino, url de vídeo ou imagem e por fim, seleciona todos os atributos que caracterizam o exercício e classifica-os numa escala de 0 a 5.

UR04 – Editar/visualizar/eliminar exercícios: O utilizador registado seleciona a opção de Exercícios no menu de aplicação, onde terá um novo painel aonde deverá indicar a ação que pretende realizar, as opções disponíveis são: para além da opção de criar novo exercício explicado no caso de uso anterior, visualizar detalhes de exercício(s), editar exercício(s) ou remover exercício(s).

UR05 – Criação manual de treinos: O utilizador registado seleciona a opção de “Criar novo treino”, onde terá que escolher o tipo de criação, optando pela criação manual de treino. Introduz os dados solicitados pela aplicação, tais como, nome, data em que o treino vai ser agendado, observações e seleciona o conjunto de exercícios que pretende que seja inserido no treino, o campo da duração do treino vai ser atualizado à medida que vai ser adicionado exercícios.

UR06 – Criação automática de treinos com base no algoritmo de recomendação: O utilizador registado seleciona a opção de “Criar novo treino”, onde terá que escolher o tipo de criação, optando pela criação com recomendação. Introduz o email do jogador que pretende obter a recomendação e o período de análise para a mesma. De seguida, insere o nome, data em que o treino vai ser agendado, observações e duração do treino em minutos, e seleciona a opção de gerar treino que com base na duração inserida e com o algoritmo implementado gera os exercícios.

UR07 – Editar/Eliminar/Visualizar detalhes do treino: O utilizador registado seleciona a opção de Treinos no menu de aplicação, onde terá um novo painel aonde deverá indicar a ação que pretende realizar, as opções disponíveis são: para além da opção de criar novo treino explicado nos dois casos de uso anteriores, visualizar detalhes de treino(s), editar treino(s) ou remover treino(s).

UR08 – Visualizar calendário com agendamento dos treinos: O utilizador registado seleciona a opção de Planos de treino no menu de aplicação, onde irá visualizar um calendário com todos os treinos agendados.

UR09 – Dashboard estatístico: O utilizador registado seleciona a opção de Dashboard no menu de aplicação, onde poderá visualizar gráficos estatísticos relativos aos exercícios, treinos e atributos.

4.2.4.3 Casos de uso do Utilizador não registado

UNR1 – Registo na aplicação: O utilizador não registado para ter acesso às funcionalidades do sistema, seleciona a opção de registo na aplicação, insere todos os dados necessários para efetuar o mesmo e aguarda pela aprovação.

4.2.5 Requisitos não funcionais

Os requisitos não funcionais, são requisitos que especificam como o sistema se deve comportar. Os mesmos, podem ser considerados como atributos do sistema.

Para definir os requisitos não funcionais, é utilizado o modelo FURPS+ (acrónimo de Functionality, Usability, Reliability, Performance e Supportability e + para categorias adicionais), este modelo, permite a fácil classificação de requisitos não abrangidos pelos requisitos funcionais.

Assim sendo, tendo em conta as entidades presentes no Modelo FURPS+, serão apresentados os principais requisitos não funcionais identificados para o sistema a desenvolver:

1. Funcionalidade:

- Autenticação dos utilizadores para aceder às funcionalidades do sistema.
- Autorização dos acessos a certas funcionalidades, tendo em conta o seu role no sistema.

- Segurança dos dados de cada utilizador do sistema, como as suas credenciais de acesso, os seus dados pessoais, entre outros.
2. Usabilidade:
- Pretende-se que o sistema seja ágil, fácil de usar e completo em termos de informação.
3. Confiabilidade:
- A aplicação deve ter uma taxa de erro muito baixa.
4. Desempenho:
- A especificação funcional deve ter em consideração o estado da arte bem como os requisitos que irão ser levantados.
5. Suportabilidade:
- A aplicação deve suportar todos os navegadores web disponíveis.
 - O sistema tem de ser facilmente testado.
 - O sistema tem de ser extensível.
 - O sistema tem de ser de fácil manutenibilidade.
6. Restrições de Design:
- Adoção do processo de desenvolvimento de software iterativo e incremental.
 - Adoção de boas práticas de design, nomeadamente padrões GRASP.
 - O sistema de base de dados a usar é MySQL.
7. Restrições de implementação:

- Adoção de normas de codificação.
- A aplicação web deve ser responsiva e acessível através das mais variadas plataformas (e.g. PC, Tablet, Smartphone).
- Todas as tecnologias usadas têm de ser *open source*.

8. Restrições de Interface:

- Interfaces simples e intuitivas

4.3 Arquitetura da Solução

Nesta secção será exposta a arquitetura da solução desenvolvida, como referido anteriormente, para descrever o software com a maior precisão possível, a estrutura utilizada é baseada no modelo de arquitetura "4 + 1", como é ilustrado na Figura 12.

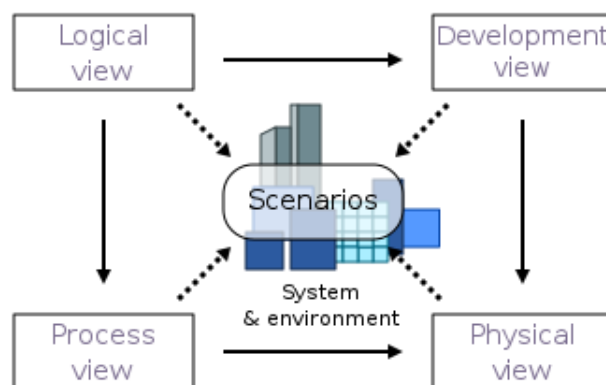


Figura 10 Modelo de Arquitetura 4+1 ²⁹

Contudo, de maneira a otimizar a arquitetura proposta pelo modelo acima, é apresentada uma nova vista referente à persistência de dados (vista de Dados), em substituição da vista de Processos, uma vez que neste projeto, a mesma, é incluída na vista de casos de uso (cenários). Assim, as vistas utilizadas para documentar o presente sistema são:

²⁹ P.B Kruchten, The 4+1 View Model of architecture. 1995

- **Vista Lógica:** Descreve todos os componentes do sistema.
- **Vista de Implantação:** Descreve quais e onde os componentes do sistema estão instalados.
- **Vista de Implementação:** Descreve as camadas da aplicação.
- **Vista de Dados:** Descreve os elementos persistentes arquiteturalmente significativos no modelo de dados.
- **Vista de Casos de Uso (Cenários):** Descreve a relação entre os atores e os casos de uso do sistema.

4.3.1 MVC (Model-View-Controller)

O padrão arquitetural Model-View-Controller (MVC) ³⁰ divide uma aplicação interativa em três camadas:

- Model – contém as funcionalidades e os dados principais.
- Views – apresenta as informações ao utilizador.
- Controllers – trata os dados do utilizador.

A combinação das views e controller dão origem á interface de utilizador (UI).

Funciona bem em conjunto com a programação orientada a objetos, visto que, os diferentes *models*, *views* e *controllers*, podem ser tratados como objetos.

A separação de camadas tem como objetivo a separação da lógica de negócio da camada de apresentação, ou seja, haverá uma divisão entre a interface de utilizador (UI) e a lógica do sistema, aumentando assim, a capacidade manutenibilidade e reutilização

³⁰ JOSUÉ LUCIANO; WALLISON JOEL BARBERÁ ALVES, PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER

do projeto. A organização em camadas é a chave para a independência entre os componentes e esta vai permitir atingir os objetivos de eficiência, escalabilidade, reutilização e manutenibilidade.

A camada “Model”, manipula os dados internos do sistema e comunica com a base de dados.

A camada “View”, é a camada responsável pela interface de utilizador, ou seja, comunica com o *controller*, o que possibilita o acesso a dados do *Model* e assim apresenta os dados conforme o evento.

Por fim, a camada “Controller”, exerce as funcionalidades que envolvem o comportamento da aplicação e controla os fluxos entre as componentes *View* e *Model*.

Estes três elementos têm baixo acoplamento e alta coesão.

O *Model*, mantém o estado da aplicação e é mais do que uma classe para armazenar dados, nele devem estar todas as regras de negócios e também, se necessário, comunicar com uma base de dados.

A *View*, específica exatamente como o model deve ser apresentado, é dinâmica, ou seja, adaptasse a qualquer modificação do model.

O *Controller*, traduz as interações do utilizador com a view, mapeando-as para tarefas que o model irá realizar.

A figura 11, exemplifica a relação entre as camadas no padrão arquitetural MVC.

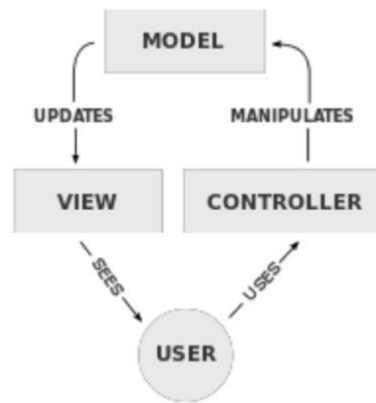


Figura 11 - Padrão Model-View-Controller³¹

4.3.2 Vista Lógica

A vista lógica identifica os principais subsistemas e packages aplicando uma separação e distribuição de responsabilidades. Nas próximas secções é mostrado, inicialmente, uma vista de alto nível e, de seguida, uma vista para cada um dos principais componentes presentes nessa vista descrevendo os mesmos.

4.3.2.1 Arquitetura Geral

A arquitetura da solução, num nível mais elevado, assenta, essencialmente nos seguintes componentes:

³¹ A.Leff, J.T.Rayfield, Web-application development using Model/View/Controller design pattern. 2001

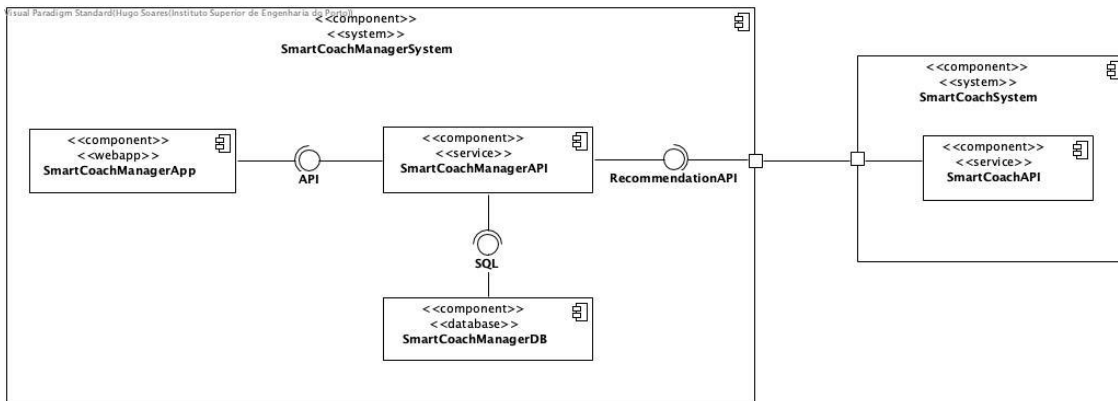


Figura 12 Diagrama de Componentes da Arquitetura Geral

- **SmartCoachManagerSystem**: Componente principal, que representa o sistema como um todo.
- **SmartCoachManagerApp**: Aplicação cliente que corre num web browser e está disponível para ser acedida a qualquer momento por qualquer utilizador. Para ter acesso aos dados, a componente comunica com o SmartCoachManagerAPI através de ligações HTTPs.
- **SmartCoachManagerAPI**: Componente responsável por disponibilizar serviços que contém a lógica de negócio da aplicação. Este componente tem a responsabilidade de comunicar com a base de dados através de SQL e de disponibilizar uma interface HTTPs e Rest facultando serviços.
- **SmartCoachManagerDB**: Componente responsável por armazenar os dados referente ao domínio implementado.
- **SmartCoachSystem**: Sistema independente que corresponde a um sistema de recomendação de planos de treino, é composto pela SmartCoachAPI e a sua Base de dados.
- **SmartCoachAPI**: Componente responsável por “consumir” os serviços do SmartCoachManagerSystem.

4.3.2.2 Arquitetura Backend

O componente SmartCoachManagerAPI representa a arquitetura backend do projeto, sendo representada na figura seguinte a estruturação interna do mesmo.

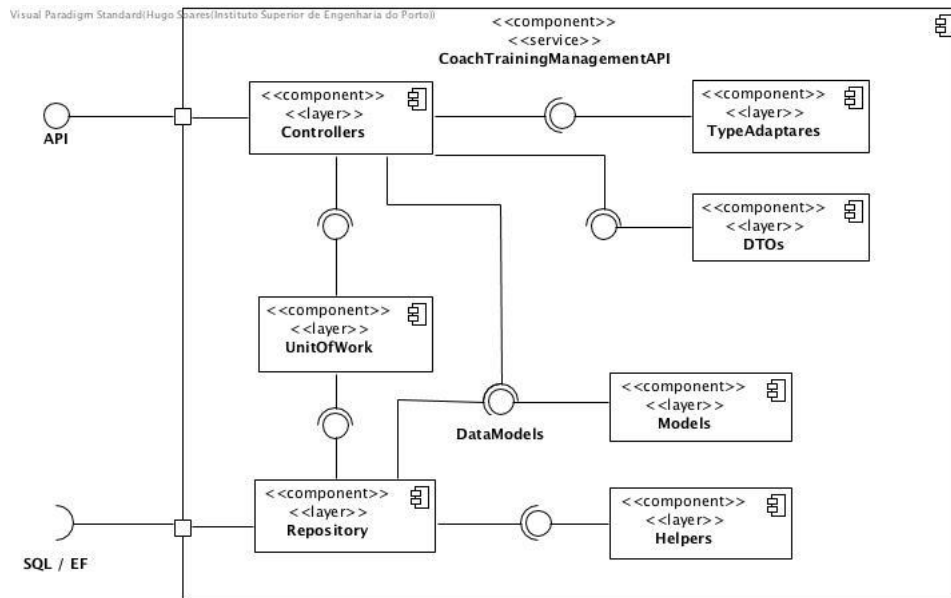


Figura 13 Diagrama de Componentes da Arquitetura Backend

Cada componente representado no diagrama anterior representa uma camada na API. De seguida, serão explicados todas os componentes representados na figura 14:

- **DTOs**: Usado para transferir dados entre a API e a aplicação agrupando um conjunto de valores (atributos) numa classe simples de forma a otimizar a comunicação. Não apresenta lógica de negócio;
- **Controller**: Responsável por tratar dos pedidos à API. Usa DTOs para a receção e envio de dados e utiliza o **UnitOfWork** para a obtenção de dados provenientes da base dados:
- **UnitOfWork**: Responsável por manter uma lista de objetos afetados por uma transação e coordenar a escrita de mudanças, tratando, caso seja necessário, de possíveis problemas de concorrência. Por outras palavras, numa ação específica

do utilizador (ex. registo), todas as transações como criar, editar, remover, entre outras, são feitas numa única transação.

- Repository: Responsável por permitir que todo o código use objetos sem ter o conhecimento como estes são persistidos, permitindo uma abstração no código. Possui todo o conhecimento da persistência, incluindo o mapeamento de tabelas para objetos;
- Model: Responsável por representar as classes de domínio e lógica de negócio sempre que necessário.
- Helpers: Responsável por possuir classes que sustentam o bom funcionamento da aplicação, como o SeedData (dados teste para inicializar base de dados), AutoMapperProfile (mapeamento entre Dtos e Models), DataContext, entre outras.
- TypeAdapters: Responsável por possuir a lógica necessária para a conversão entre objetos DTOs e Models, assim como, a operação inversa.

4.3.2.3 Arquitetura Frontend

O componente SmartCoachManagerApp representa a arquitetura frontend do projeto, optou-se por utilizar o padrão MVC, como tal a figura seguinte representa a estruturação interna do mesmo.

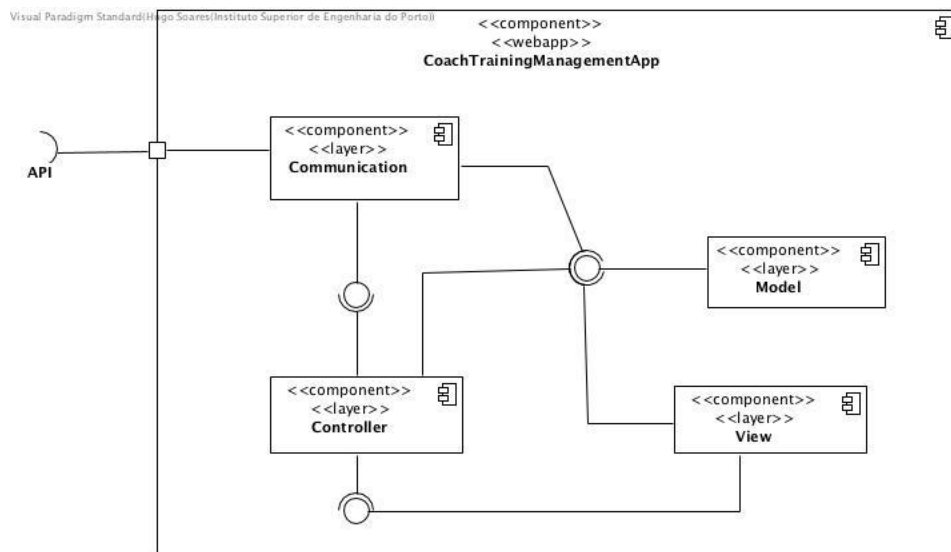


Figura 14 Diagrama de Componentes da Arquitetura Frontend

Os componentes presentes na figura anterior são:

- **Controller**: Responsável por possuir métodos de ação que lidam com as solicitações do browser, recuperar dados necessários do model e retornar as respostas apropriadas.
- **Model**: Responsável por responder a pedidos, com o intuito de devolver informação sobre o estado da mesma (normalmente oriundos da view) e responder a instruções para mudar o estado da informação (normalmente oriundos do controller).
- **View**: Responsável por gerir a apresentação de dados da aplicação e a interação com o utilizador.
- **Communication**: Responsável por tratar da comunicação com a API, permitindo o desacoplamento da camada de tratamento de eventos da camada de comunicação com serviços externos.

4.3.3 Vista de Implantação

A vista de implantação é um diagrama que apresenta os nós em execução e os componentes neles inseridos. O diagrama da Figura 26 apresenta a vista de implantação da aplicação “SmartCoachManager”.

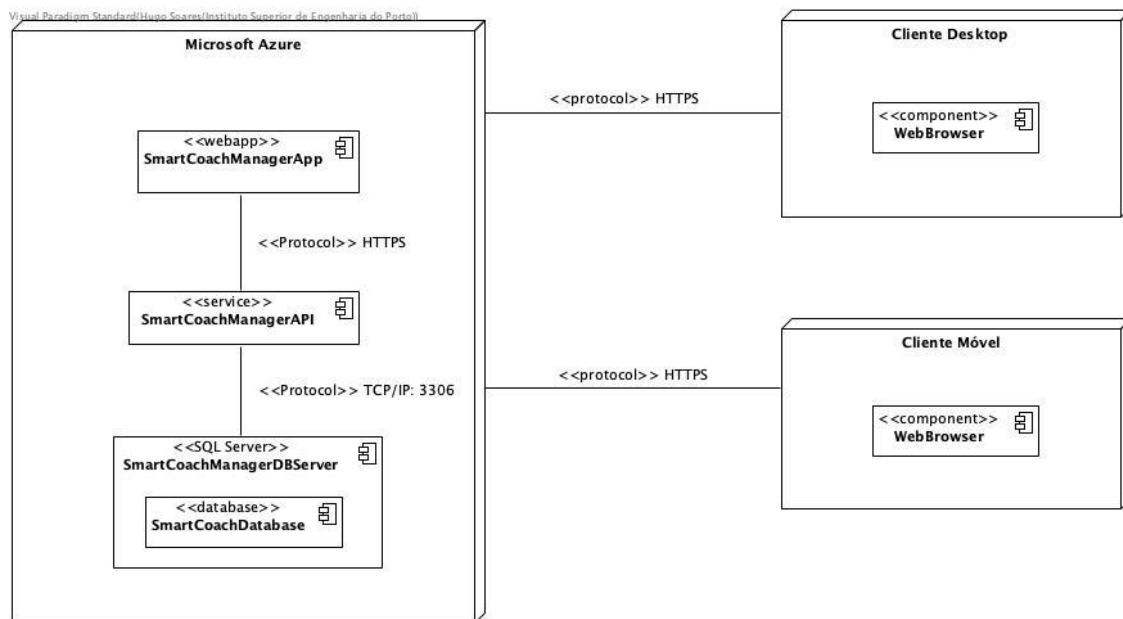


Figura 15 - Vista de Implantação

Analisando o diagrama, verificamos a existência do seguinte nó:

- **Microsoft Azure** – representa a instalação da aplicação “SmartCoachManager” no servidor Microsoft Azure, ficando assim disponível online. A aplicação web “SmartCoachManagerApp” e a API “SmartCoachManagerAPI” executam as duas no servidor Azure, e comunicam entre si através do protocolo HTTPS. Estas comunicam com uma base de dados MySQL, através da porta 3306. Os utilizadores acedem à aplicação web utilizando um *browser (webbrowser)*.

4.3.4 Vista de Implementação

A vista de Implementação é um diagrama que descreve a implementação física de informações geradas pelo programa de *software* em componentes de *hardware*. Representa a configuração e a arquitetura de um sistema.

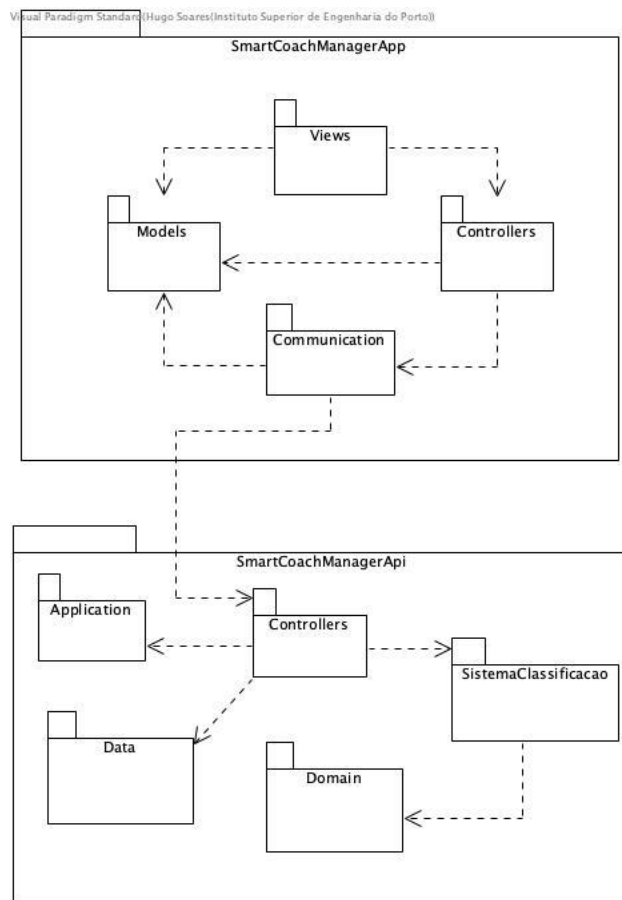


Figura 16 – Vista de Implementação

Analisando a figura 15, verificamos a existência de dois módulos:

- **SmartCoachManagerApp** – representa a aplicação web. É este componente que recebe os pedidos do *browser*, sendo o principal ponto de entrada do sistema. A lógica da UI encontra-se implementada neste ponto. Neste componente está representado as views, controllers da camada de apresentação.
- **SmartCoachManagerAPI** – representa a API da aplicação. É este componente que recebe os pedidos provenientes de outras instâncias da aplicação. Neste componente está representado, os services, DTO's e o sistema de classificação de treinos e exercícios.

4.3.5 Vista de Dados

A vista de dados é um diagrama que pretende representar o modelo da base de dados. Visto que este diagrama tem alguma dimensão, decidiu-se colocá-lo em anexo (Anexo 1).

4.3.6 Vista de Casos de Uso (Cenários)

Tendo em consideração os casos de uso identificados na **secção 4.2.4.1**, e como o âmbito deste documento não é a especificação de requisitos, apenas serão detalhados os casos de uso considerados mais relevantes. Os casos de uso selecionados foram o UNR1 registar na aplicação, UR3 criar novo exercício por ser um dos requisitos principais da aplicação, que permite classificar atributos quanto ao exercício que está a ser criado, UR5 e UR6, criar treino manual e com recomendação, respetivamente.

4.3.6.1 UNR1: Registar na aplicação

Tabela 5 – Especificação do caso de uso “UNR1: Registar na aplicação”.

Código: UNR1
Ator: Utilizador não registado
Pré-condições: <ul style="list-style-type: none">• Utilizador não está registado na aplicação.
Pós-condições: <ul style="list-style-type: none">• Utilizador encontra-se registado na aplicação.
Cenário principal de sucesso (fluxo básico): <ol style="list-style-type: none">1. O utilizador não registado inicia o processo de registo na aplicação.2. O sistema solicita os dados necessários para o registo na mesma (e-mail, <i>password</i>).3. O utilizador introduz, no sistema, os dados solicitados pelo mesmo.4. O sistema valida a informação.5. O sistema regista o novo utilizador.
Extensões (fluxos alternativos): <ol style="list-style-type: none">4a. O sistema deteta informação inválida.<ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro.
Requisitos especiais: <ul style="list-style-type: none">• A password tem de cumprir o seguinte formato: mínimo de 6 caracteres e pelo menos uma letra maiúscula.• Todos os campos são de preenchimento obrigatório.

4.3.6.2 UR3: Criar novo exercício

Tabela 6 - Especificação do caso de uso "UR3: Criar novo exercício".

Código: UR3
Ator: Utilizador registado
Pré-condições: <ul style="list-style-type: none">• Utilizador tem a sessão iniciada no sistema.
Pós-condições: <ul style="list-style-type: none">• O novo exercício contém os atributos associados e a sua classificação.• O novo exercício fica armazenado na base de dados.
Cenário principal de sucesso (fluxo básico): <ol style="list-style-type: none">1. O utilizador inicia o processo de criação de um novo exercício.2. O sistema solicita os dados necessários (nome, descrição, duração em minutos, url de vídeo ou imagem, seleccionar os atributos e classificá-los num intervalo de 0 a 5 estrelas).3. O utilizador introduz no sistema, os dados solicitados pelo mesmo.4. O sistema valida a informação.5. O sistema armazena na sua base de dados o novo exercício.
Extensões (fluxos alternativos): <ol style="list-style-type: none">4a. O sistema deteta informação inválida.<ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro.4b. O sistema deteta que não foram seleccionados quaisquer atributos.<ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro.4c. O sistema deteta dados obrigatórios não preenchidos.<ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro.
Requisitos especiais: <ul style="list-style-type: none">• O atributo tem que estar classificado numa escala correspondida de 0 a 5.• Pode ser seleccionado mais do que um atributo.• Os campos "nome", "descrição", "duração" e "atributos" são de preenchimento obrigatório.

4.3.6.3 UR5: Criar treino manual

Tabela 7 - Especificação do caso de uso "UR5: Criar treino manual".

Código: UR5
Ator: Utilizador registado
Pré-condições: <ul style="list-style-type: none">• Utilizador tem a sessão iniciada no sistema.
Pós-condições: <ul style="list-style-type: none">• O novo treino fica com exercícios associados.• O novo treino fica armazenado na base de dados.

<p>Cenário principal de sucesso (fluxo básico):</p> <ol style="list-style-type: none"> 1. O utilizador inicia o processo de criação de um novo treino manual. 2. O sistema solicita os dados necessários (nome, data, observações, seleccionar os exercícios pretendidos). 3. O utilizador introduz no sistema, os dados solicitados pelo mesmo. 4. O sistema valida a informação. 5. O sistema armazena na sua base de dados o novo treino.
<p>Extensões (fluxos alternativos):</p> <ol style="list-style-type: none"> 4a. O sistema deteta dados obrigatórios não preenchidos. <ul style="list-style-type: none"> • O sistema informa o utilizador através de uma mensagem de erro. 4b. O sistema deteta informação inválida. <ul style="list-style-type: none"> • O sistema informa o utilizador através de uma mensagem de erro.
<p>Requisitos especiais:</p> <ul style="list-style-type: none"> • A data segue o formato (dd/mm/aaaa, Hora: minutos). • Pode ser seleccionado mais de que um exercício. • Os campos “nome”, “data”, “observações” e “exercícios” são de preenchimento obrigatório.

4.3.6.4 UR6 - Criar novo treino com recomendação

Tabela 8 - Especificação do caso de uso “UR6 - Criar novo treino com recomendação”.

Código: UR6
Ator: Utilizador registado
<p>Pré-condições:</p> <ul style="list-style-type: none"> • Utilizador tem a sessão iniciada no sistema. • A aplicação tem de estar conectada a um sistema de recomendação externo.
<p>Pós-condições:</p> <ul style="list-style-type: none"> • O novo treino fica com exercícios associados. • O novo treino fica armazenado na base de dados.
<p>Cenário principal de sucesso (fluxo básico):</p> <ol style="list-style-type: none"> 1. O utilizador inicia o processo de criação de um novo treino com recomendação. 2. O sistema solicita o e-mail do jogador e o respetivo período de análise. 3. O utilizador introduz os respetivos dados. 4. O sistema valida a informação. 5. O sistema solicita os dados necessários (nome, data, duração em minutos, observações). 6. O utilizador fornece os dados solicitados e selecciona a opção de gerar exercícios. 7. O sistema valida a informação. 8. O sistema armazena na sua base de dados o novo treino.
<p>Extensões (fluxos alternativos):</p> <ol style="list-style-type: none"> 4a. O sistema deteta e-mail inválido.

<ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro. <p>4b. O sistema verifica que para o período de análise inserido não é possível gerar recomendação.</p> <ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro. <p>7a. O sistema deteta dados obrigatórios não preenchidos.</p> <ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro. <p>7b. O sistema deteta informação inválida.</p> <ul style="list-style-type: none">• O sistema informa o utilizador através de uma mensagem de erro.
Requisitos especiais: <ul style="list-style-type: none">• A data segue o formato (dd/mm/aaaa, Hora: minutos).• Os campos “nome”, “data” e “duração” são de preenchimento obrigatório.

5 Implementação

Neste capítulo serão apresentados os detalhes de desenvolvimento da solução anteriormente apresentada, focando-se essencialmente no detalhe da aplicação web e no sistema de classificação e recomendação. Os diagramas apresentados neste capítulo têm uma granularidade mais fina, visto estarem mais próximos da implementação efetuada.

5.1 Aplicação SmartCoachManager

Na subsecção 4.3.1.3 foi apresentado o estilo arquitetural MVC como base a seguir para a implementação do projeto. A figura 16 apresenta a estrutura de pastas vista na perspetiva do IDE utilizado para o desenvolvimento, o Visual Studio for Mac 2019.

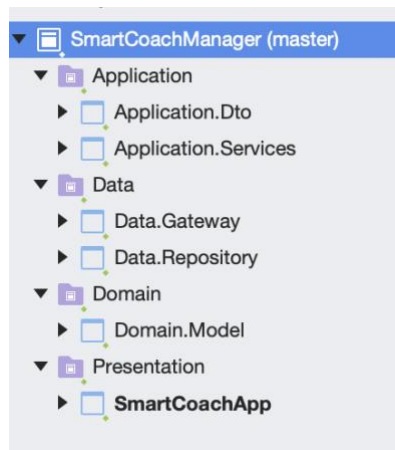


Figura 17 - Organização do projeto

As subsecções seguintes apresentam o detalhe de cada uma das camadas da arquitetura da aplicação.

5.1.1 Domain

A componente “Domain” contém toda a lógica do negócio e os respetivos modelos responsáveis por armazenar os dados e dos serviços de domínio. Foi seguida uma abordagem de desenvolvimento orientada ao domínio, pelo que, a lógica de negócio está presente nas entidades de domínio. O diagrama da Figura 17, apresenta a vista de módulos da componente.

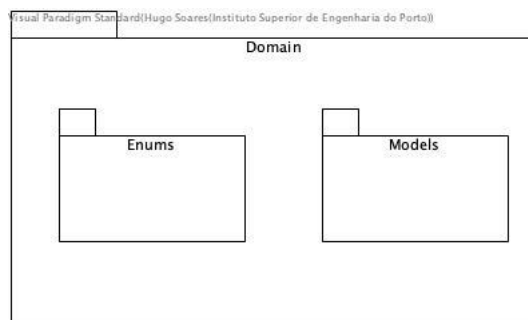


Figura 18 - Vista de módulos da componente Domain

Sendo o módulo de um projeto que segue o padrão arquitetural MVC, é de esperar encontrar referências a “Models”.

5.1.2 Application

A componente de “Application”, juntamente com a componente anterior são consideradas o centro da aplicação e não devem possuir nenhuma dependência para além da própria Framework em que estão inseridas. Contém os DTO's (*Data transfer objects*) e os serviços responsáveis para o acesso a dados. O diagrama da Figura 18, apresenta a vista de módulos da componente.

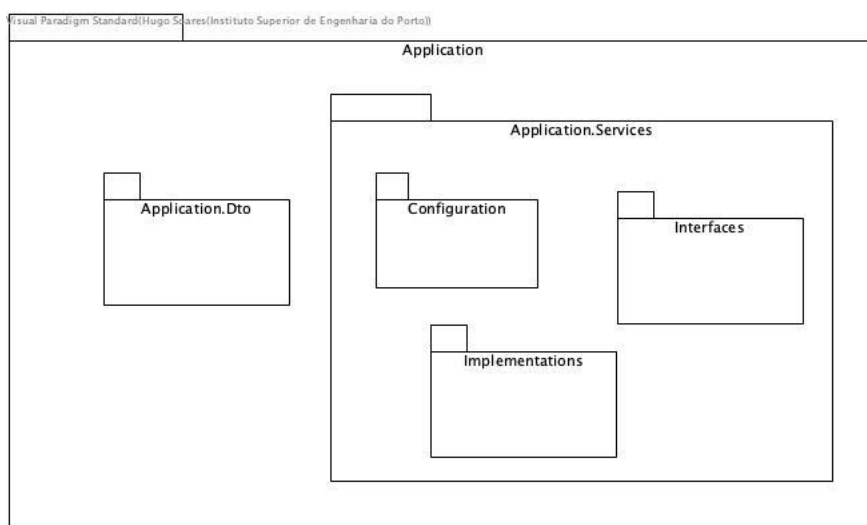


Figura 19 - Vista de módulos da componente Application

O diagrama da Figura 18, apresenta os seguintes módulos:

- Application.Dto: Contém a definição dos DTO's utilizados na aplicação;
- Application.Services: Contém todas a definição de todas as interfaces, services e configurações para serviços externos ou de acesso a dados. Estas interfaces incluem abstrações para operações realizadas pela camada de Data, tais como o acesso a dados ou a serviços externos;

5.1.3 Data

A componente “Data”, contém a lógica de acesso a dados ou a serviços externos. O diagrama da Figura 19, apresenta a vista de módulos da componente.

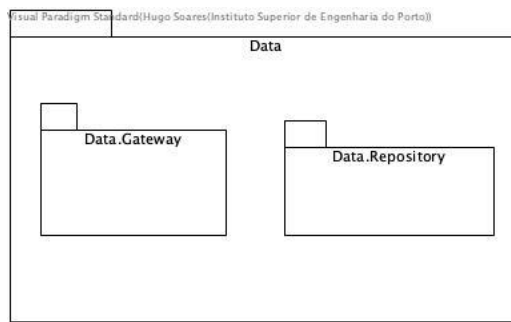


Figura 20 - Vista de módulos da componente Data

O diagrama da Figura 19, apresenta os seguintes módulos:

- Data.Gateway: Contém a lógica de acesso a serviços externos, tais como o sistema de recomendação.
- Data.Repository: Contém a lógica de acesso a dados. Este módulo utiliza o *Entity Framework Core* como ORM.

5.1.4 Presentation

A componente “Presentation”, contém toda a lógica de apresentação da aplicação. É a ligação gráfica entre o utilizador e a aplicação. O diagrama da Figura 20 apresenta a vista de módulos da componente.

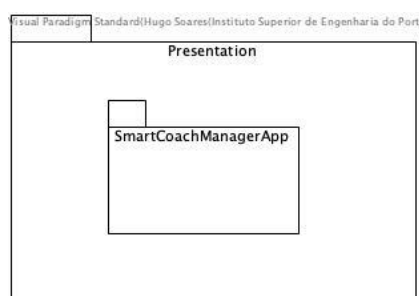


Figura 21 - Vista de módulos da componente Presentation

O diagrama de Figura 20, apresenta o seguinte módulo:

- SmartCoachManagerApp: representa a aplicação web utilizada pelos utilizadores. Trata-se de um projeto Asp.Net Core Razor Pages, que segue o

padrão arquitetural MVC, com a vantagem de encapsular os *controllers* e as *views* diretamente numa *page*.

5.1.5 Evidências de boas práticas e do uso de padrões de *software*

O uso de boas práticas e de padrões de *design* de software foram as duas maiores considerações no decorrer deste projeto. Em seguida, serão demonstrados alguns exemplos de casos concretos onde essas considerações são visíveis.

5.1.5.1 Padrão Repositório + *Unit of Work*

O padrão de *design* de Repositório³² em C# a partir do contexto Entity Framework das aplicações ASP.NET.MVC é um dos padrões de design mais utilizados. Isto porque, atualmente, grande parte das aplicações controladas por dados precisam de aceder a dados que residem numa ou mais fontes, como tal, a forma mais simples é escrever/criar um código que esteja relacionado com o acesso aos dados na aplicação principal.

Este padrão intervém entre o domínio e as camadas de mapeamento de dados, utilizando uma *interface* semelhante a uma coleção, para aceder aos objetos de domínio, por outras palavras, podemos afirmar que este padrão atua como intermediário entre a aplicação e a lógica de acesso a dados.

Sendo assim, o padrão de repositório isola todo o código de acesso aos restantes dados da aplicação, sendo que, uma das grandes vantagens deste é que se for necessário realizar alguma alteração, esta só será realizada num lugar.

Um dos aspetos mais importantes deste padrão é a separação entre a base de dados atual, as consultas e outras lógicas de acesso a restantes dados da aplicação.

Este padrão foi implementado no package SmartCoachManager.Data. A figura 22, apresenta o exemplo da implementação.

³² Scott Millett, Nick Tune, Patterns, Principles and Practices of Domain-Driven Design

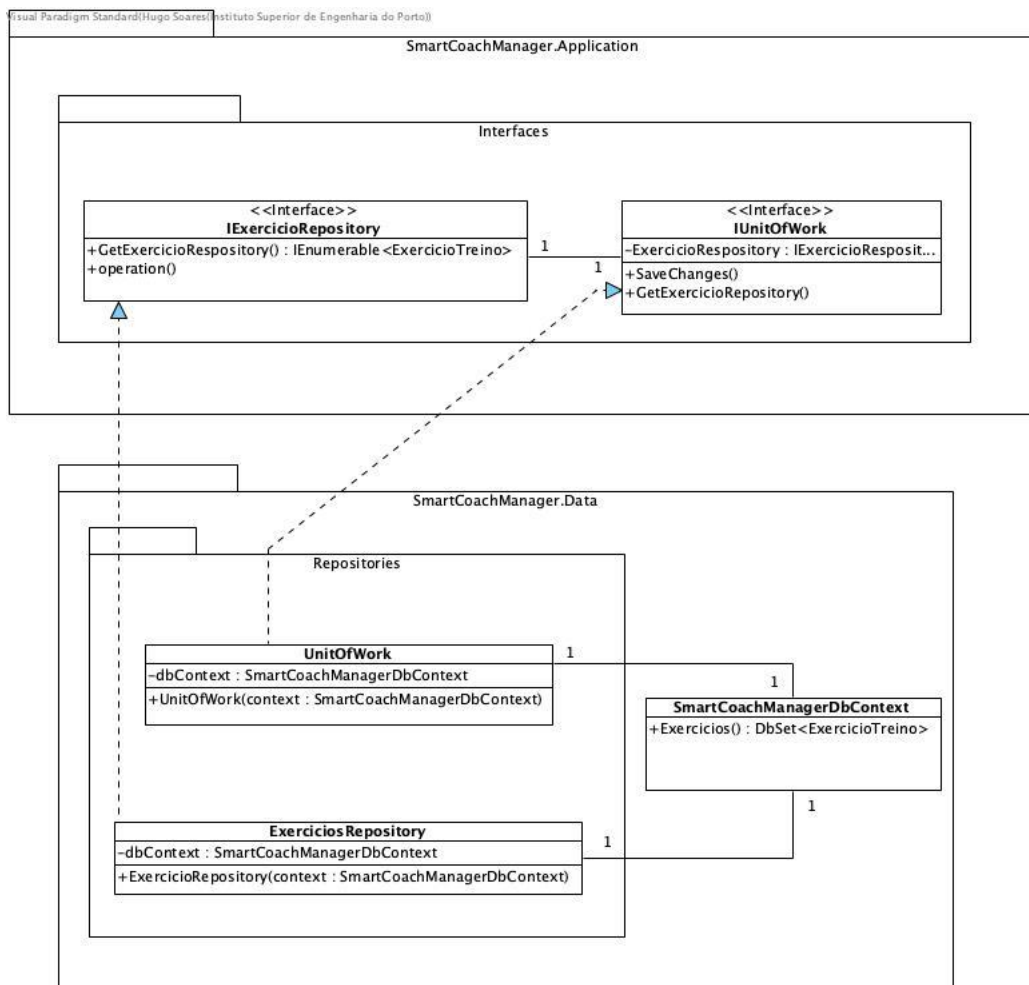


Figura 22 - Implementação do padrão Repositório

No diagrama é possível constatar a existência da classe “SmartCoachManagerDbContext”. Esta classe é específica do Entity Framework Core e representa uma sessão com a base de dados que pode ser utilizada para realizar consultas ou alterações.

5.2 Casos de Uso

Nesta secção serão apresentadas as funcionalidades implementadas, em casos específicos, irá ser realizada uma análise mais detalhada que inclui a apresentação de excertos de código e diagramas de sequência. Esta secção apresentará também o resultado final dos respetivos casos de uso.

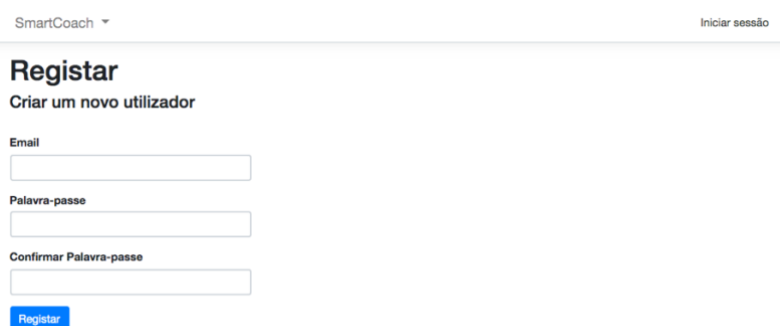
5.2.1 Login/Registo na aplicação

Para usufruir das utilidades da aplicação, o utilizador tem de criar um novo registo na aplicação e fazer o respetivo login.

Para a concretização destes dois casos de uso, foi instituído como sistema de autenticação o ASP.NET Identity que utiliza o *Entity Framework Core* para comunicar com a base de dados. EF Core usa o migrations para manter a base de dados sincronizada com o *Model*.

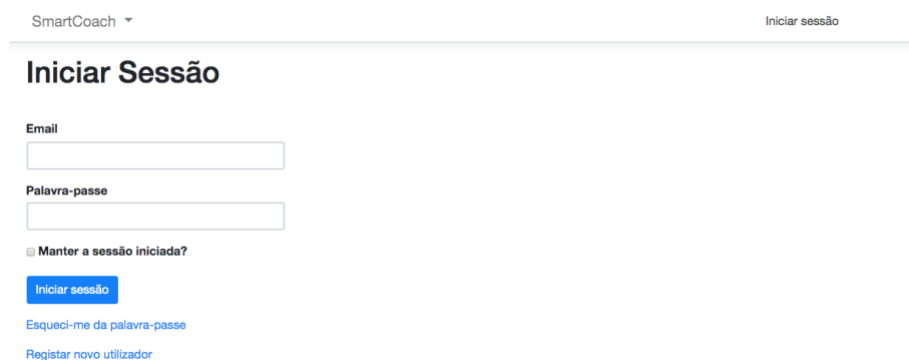
No decorrer da criação do projeto no Visual Studio, foi definido o *Identity* como sistema de autenticação e automaticamente foi gerada uma pasta denominada *Account*. Esta pasta, contém Razor Pages que tem a função de gerir as tarefas relacionadas com a autenticação.

Nas Figuras 22 e 23, está representado os respetivos *layouts* da aplicação.



The screenshot shows the registration page of the SmartCoach application. At the top left, there is a logo 'SmartCoach' with a dropdown arrow. At the top right, there is a link 'Iniciar sessão'. The main heading is 'Registar' in bold, followed by the sub-heading 'Criar um novo utilizador'. Below this, there are three input fields: 'Email', 'Palavra-passe', and 'Confirmar Palavra-passe'. Each field has a small icon on the right side. At the bottom of the form, there is a blue button labeled 'Registar'.

Figura 23 – Registo na aplicação



The screenshot shows the login page of the SmartCoach application. At the top left, there is a logo 'SmartCoach' with a dropdown arrow. At the top right, there is a link 'Iniciar sessão'. The main heading is 'Iniciar Sessão'. Below this, there are two input fields: 'Email' and 'Palavra-passe'. Each field has a small icon on the right side. Below the input fields, there is a checkbox labeled 'Manter a sessão iniciada?'. At the bottom of the form, there is a blue button labeled 'Iniciar sessão'. Below the button, there are two links: 'Esqueci-me da palavra-passe' and 'Registar novo utilizador'.

Figura 24 – Iniciar sessão na aplicação

5.2.2 Gestão de Atributos

Na gestão dos atributos, o utilizador tem as seguintes funcionalidades de: criar, editar, eliminar e visualizar os detalhes do mesmo.

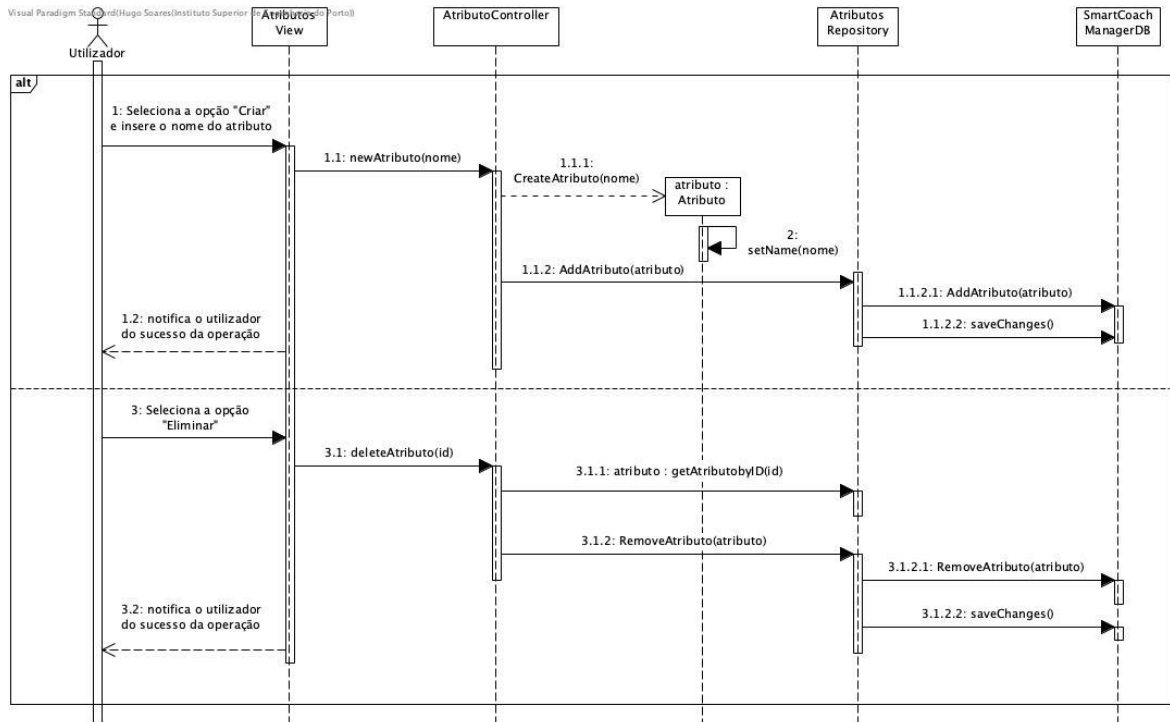


Figura 25 – Diagrama de Sequência: Gestão de atributos

Na Figura 24, está representado o diagrama de sequência da gestão de atributos, onde é detalhado as interações entre os objetos do sistema. Neste caso, a *View* dos Atributos irá ter 4 opções: criar, editar, eliminar e visualizar detalhes.

No diagrama anterior, foram especificadas as opções de criar e editar atributo. O utilizador opta por uma dessas opções e a *View* vai comunicar com o *Controller*, que, posteriormente, irá comunicar com o *Repository* para armazenar ou modificar dados em relação ao *Model* Atributo.

O *Model* Atributo é constituído pelo nome do atributo, descrição (opcional), uma referência ao utilizador que o criou e uma coleção de exercícios.

```

public class Atributo
{
    [Display(Name = "Número")]
    public int Id { get; set; }

    [Display(Name = "Nome")]
    public string Name { get; set; }

    [Display(Name = "Descrição")]
    public string Description { get; set; }

    public virtual ICollection<ExercicioTreino> ExerciciosTreino { get; set; }

    public String UserId { get; set; }
}

```

Código 1 - Model Atributo

A Figura 25, apresenta o design final da gestão de atributos na aplicação.

SmartCoach Olá, admin@admin.pt! Terminar sessão

Índice

[Criar novo atributo](#)

Nome	
Defesa Remates Curtos (GR)	Editar Eliminar Detalhes
Remate de Longe	Editar Eliminar Detalhes
Finalização	Editar Eliminar Detalhes
Posicionamento	Editar Eliminar Detalhes
Penaltie	Editar Eliminar Detalhes
Livre Direto	Editar Eliminar Detalhes
Bola Parada	Editar Eliminar Detalhes
Agilidade	Editar Eliminar Detalhes
Impulsão	Editar Eliminar Detalhes
Velocidade	Editar Eliminar Detalhes
Aceleração	Editar Eliminar Detalhes
Resistência	Editar Eliminar Detalhes
Força	Editar Eliminar Detalhes
Pé Contrário	Editar Eliminar Detalhes
Pressão Defensiva	Editar Eliminar Detalhes
Agressividade	Editar Eliminar Detalhes
Cabeceamento	Editar Eliminar Detalhes
Equilíbrio	Editar Eliminar Detalhes
Cruzamento	Editar Eliminar Detalhes
Drible	Editar Eliminar Detalhes

Figura 26 – Gestão de atributos

5.2.3 Criação de exercícios e classificação de atributos

Nesta funcionalidade o utilizador tem a função de criar exercícios, selecionar os atributos pretendidos e atribuir-lhes uma determinada classificação, num intervalo, pré-determinado, de zero a cinco (0 a 5).

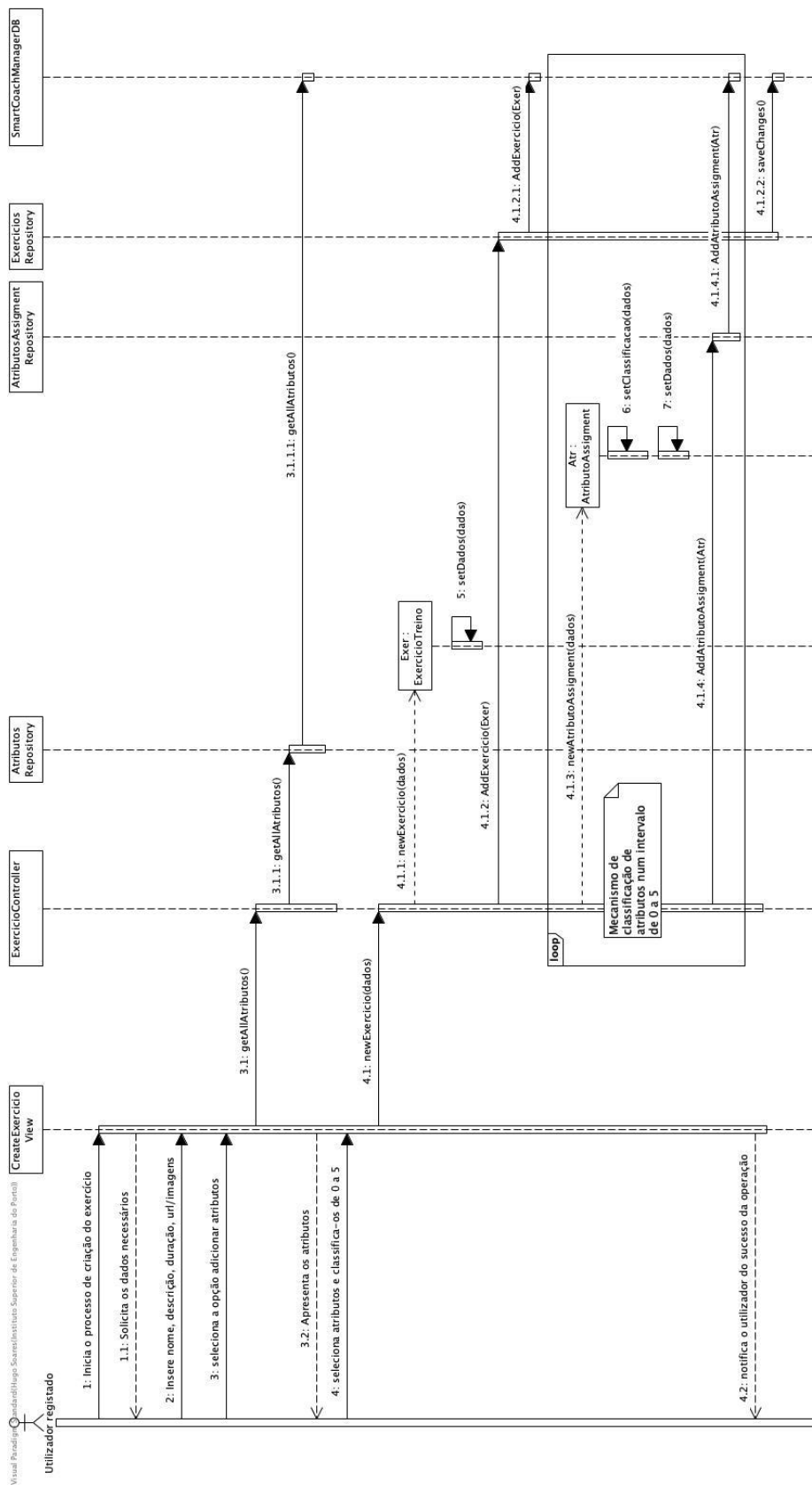


Figura 27 – Diagrama de Sequência: Criar Exercício e Classificar

Na Figura 26, está representado o diagrama de sequência do criar exercício e o respetivo sistema de classificação de atributos.

Analisando o diagrama anterior, verifica-se que o mecanismo de classificação de atributos é ativo todas as vezes que é adicionado um atributo ao exercício, ou seja, o utilizador sempre que insere um atributo no exercício, tem de classificar o mesmo numa escala de zero a cinco (0 a 5) estrelas, como demonstram as Figuras 28 e 29.

De maneira a implementar este caso de uso, seguiu-se a lógica explicada na figura 26, ou seja, foi desenvolvida a respetiva View com auxílio do *plugin* rating-star-svg, para permitir a classificação através de estrelas. Na secção 5.3, irá ser explicado o mecanismo de classificação mais detalhadamente.

The screenshot shows a web form titled 'Criar Exercício' in the SmartCoach system. The form includes several input fields: 'Nome', 'Descrição', 'Duração (Min)', and 'Video (Url)'. Below these is an 'Imagens' section with a file selection button 'Escolher Ficheiros' and a status 'Nenhum ...ionado'. At the bottom, there is an 'Atributos' section with a table header containing 'Nome', 'Classificação', and 'Opções'. A blue 'Criar' button is positioned below the table, and a 'Voltar' link is at the very bottom. The top of the page shows the user is logged in as 'admin@admin.pt!' and can 'Terminar sessão'.

Figura 28 - Criar exercício

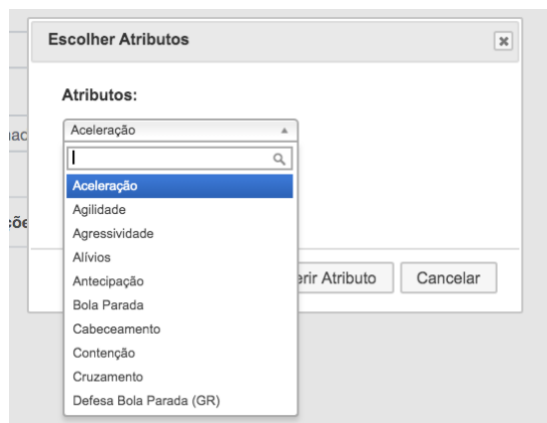


Figura 29 - Selecionar atributos

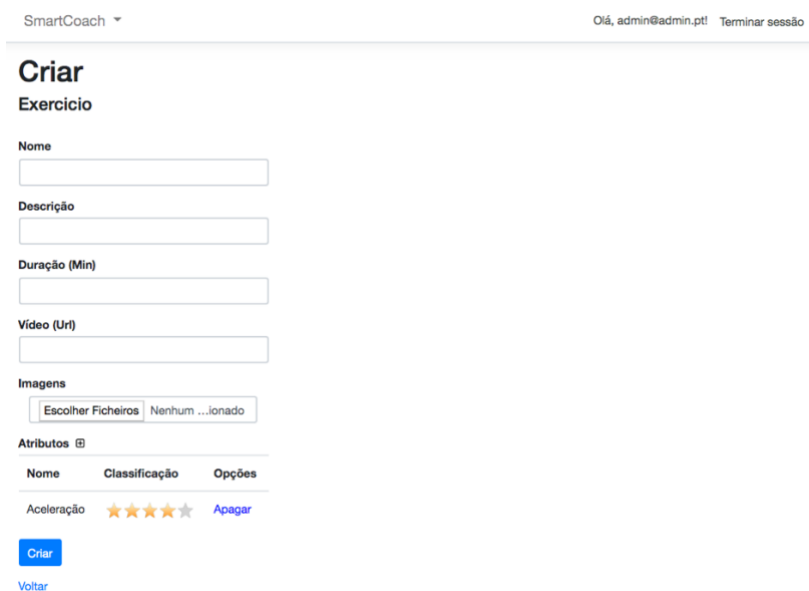


Figura 30 - Classificar os atributos

5.2.4 Editar/Visualizar/Eliminar exercícios

Esta função serve para que se o utilizador pretender tenha a capacidade de editar, visualizar e eliminar os exercícios.

Este caso de uso contém três possíveis ações do utilizador em relação aos exercícios, editar exercício, eliminar e visualizar os detalhes do mesmo.

Detalhes

Exercício

Nome 2x2+1 com cruzamento e finalização
Descrição 1) - Posse de bola 2x2+1 no meio com saída para os corredores com 1x1, cruzamento e finalização 2) - Após 5 passes com sucesso, jogar no colega que está no corredor 3) - Faz 1x1 dentro do quadrado e o que ganhar esse duelo faz cruzamento. Não pode ser perseguido depois de passar a linha 4) - Jogadores no meio atacam 1º poste e entrada da área. Jogador do corredor contrário ataca 2º poste 5) - Manter a rotação dos jogadores por posição
Duração (Min) 20
Imagens



Atributos:

Nome	Classificação
Finalização	★★★★☆
Cabeceamento	★★★★☆
Cruzamento	★★★★☆
Passo Curto	★★★★☆
Drible	★★★★☆
Recepção	★★★★☆

[Editar](#) | [Voltar](#)

Figura 31 - Visualizar detalhes do treino

Por fim, o caso de uso de eliminar é idêntico ao explicado anteriormente, diferenciando-se por conter a opção de eliminar o exercício. Quando o utilizador seleciona a opção de eliminar exercício, a view irá efetuar um pedido ao controller, e este consequentemente irá comunicar com o repositório de exercícios de maneira a remover o mesmo da base de dados.

5.2.5 Criação manual de treinos

- Nesta funcionalidade, o utilizador usufrui da capacidade de criar um treino manualmente.

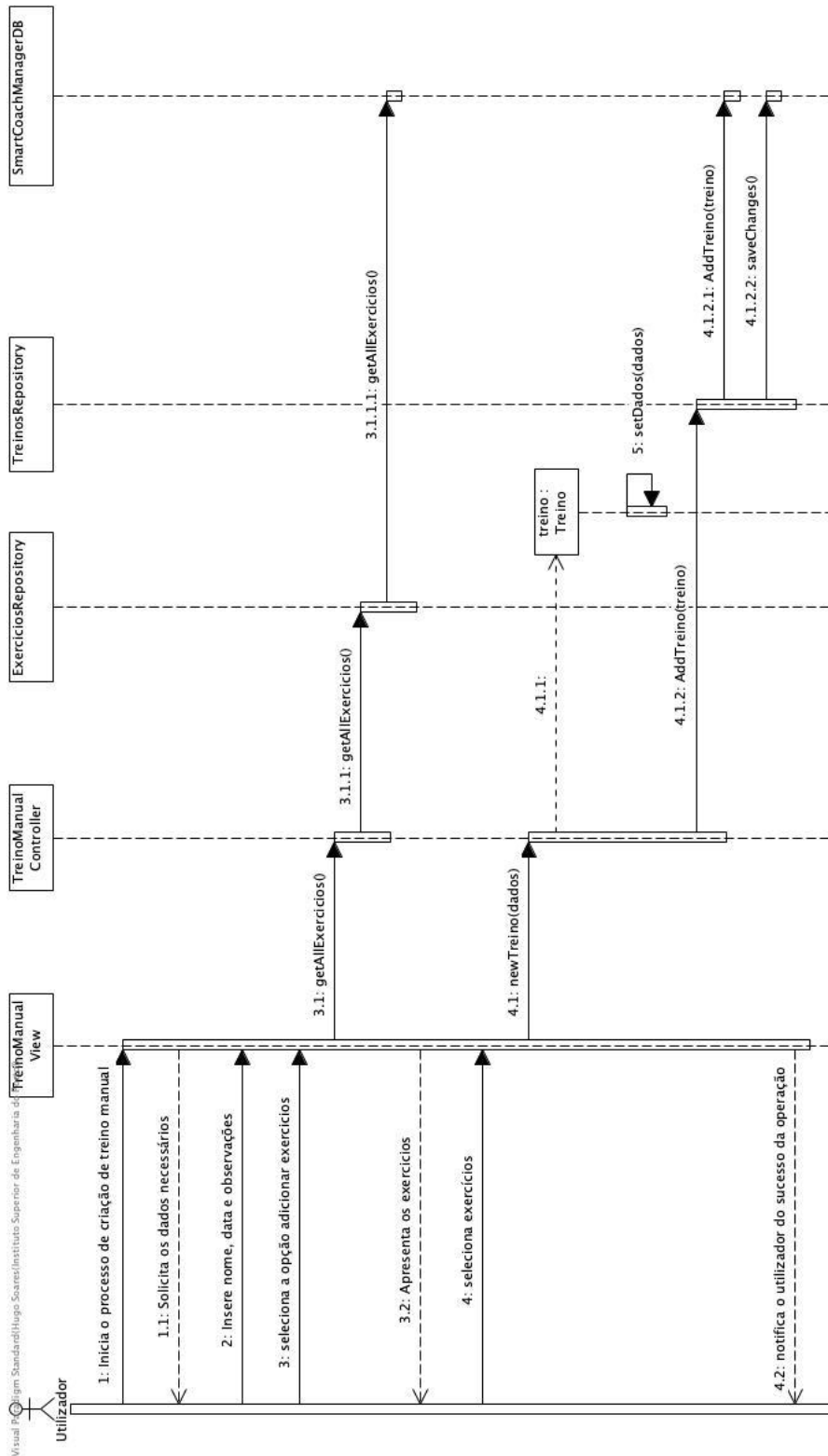


Figura 32 - Diagrama de Sequência: Criar treino manual

A criação de treinos é uma das principais funcionalidades da aplicação.

O utilizador tem a possibilidade de optar por dois tipos de criação de treino, sendo eles: treino manual ou treino com recomendação.

The screenshot shows the 'Criar Treino' (Create Training) form in the SmartCoach application. The form includes the following fields and elements:

- SmartCoach** logo and user information: 'Olá, admin@admin.pt! Terminar sessão'.
- Criar Treino** header.
- Nome**: A text input field.
- Data**: A date input field with the format 'dd/mm/aaaa, --:--'.
- Duração (Min)**: A numeric input field with the value '0'.
- Observações**: A text area for notes.
- Exercícios**: A section with a table header containing 'Nome', 'Duração (Min)', and 'Opções'.
- Criar**: A blue button to submit the form.
- Voltar**: A blue link to return to the previous screen.

Figura 33 - Criar o treino manual

Nesta subsecção, o tipo de treino abordado é a criação do treino manual, funcionalidade esta que permite que o utilizador crie de forma dinâmica um treino, adicionando os exercícios que pretende, como demonstra a figura 33.

The screenshot shows the 'Escolher Exercícios' (Choose Exercises) dialog box. It features a search bar and a list of exercises with their durations:

- 2x2+1 com cruzamento e finaliz... (20 min) - Selected
- 2x2+1+2 (15 min)
- 9v9 (20 min)
- Finalização 2x2+8J (15 min)
- Jogo reduzido 6x6+4J (20 min)
- Jogo reduzido 8x8 com 4 portas (25 min)
- MPB 7x3 (15 min)
- MPB 8x4+2J (20 min)
- Padronizado 6x0 (20 min)

Buttons for 'Adicionar Exercício' and 'Cancelar' are visible at the bottom right of the dialog.

Figura 34 - Selecionar exercícios

Para desenvolver esta funcionalidade, a *view* comunica com o *controller*, de maneira a obter todos os exercícios existentes na base de dados e assim apresentá-los ao utilizador, para que ele possa seleccionar os exercícios que pretende que o treino tenha. Sempre que é adicionado um exercício, a duração do treino é incrementada com a duração do exercício que foi introduzido, como demonstra a figura 34.

Uma das grandes vantagens desta aplicação e funcionalidade da mesma, é a partilha de conhecimento entre treinadores, havendo a possibilidade de o treinador utilizar treinos criados por outros.

SmartCoach Olá, admin@admin.pt! [Terminar sessão](#)

Criar Treino

Nome

Data

Duração (Min)

Observações

Exercícios ⊞

Nome	Duração (Min)	Opções
2x2+1+2 (15 min)	15	Apagar
Jogo reduzido 8x8 com 4 portas (25 min)	25	Apagar

[Criar](#)

[Voltar](#)

Figura 35 - Treino com exercícios

5.2.6 Criação automática de treinos com base no algoritmo de recomendação

Com esta função o utilizador dispõe de um treino automaticamente gerado pela aplicação, contudo, o utilizador necessita de inserir o e-mail do jogador e o respetivo período de análise para obter uma recomendação e o treino desejado.

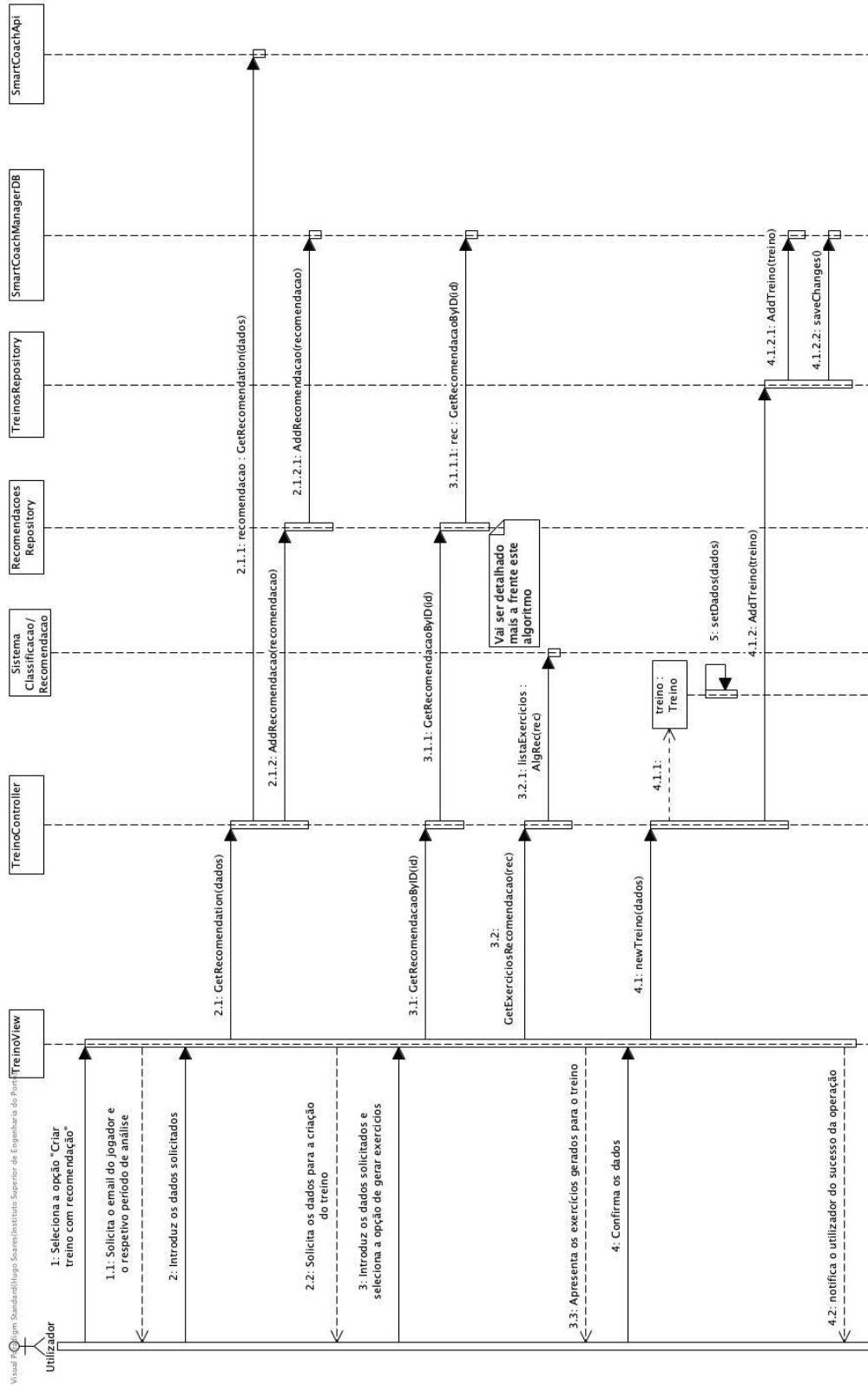
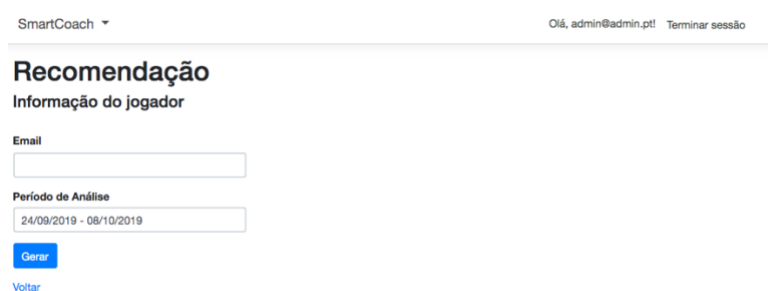


Figura 36 - Diagrama de Sequência: Criar treino com recomendação

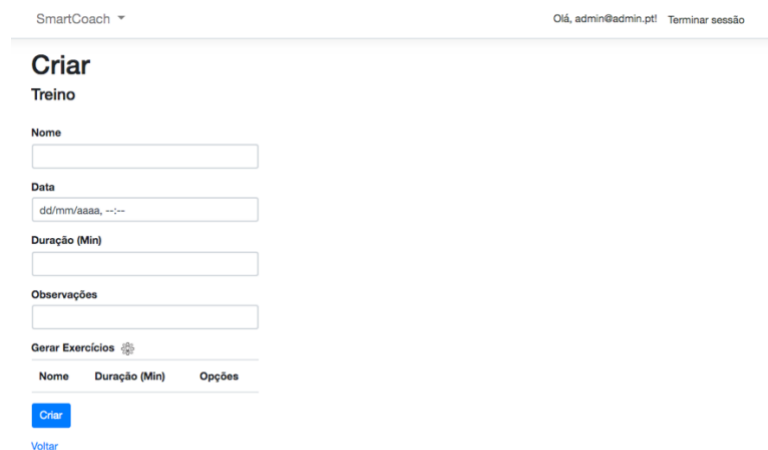
Com explicado na subsecção anterior, existem dois tipos de criação. Nesta subsecção irá ser abordado a criação de treino com recomendação.

O utilizador optando pela o tipo de criação de treino com recomendação, terá que inserir o email do jogador que pretende obter a recomendação e o período de análise do mesmo, como é demonstrado na figura 36. O sistema irá comunicar com uma api externa denominada de SmartCoachApi, em que irá obter a recomendação com os dados fornecidos pelo utilizador.



The screenshot shows the 'Recomendação' (Recommendation) page in the SmartCoach application. The page header includes 'SmartCoach' and user information 'Olá, admin@admin.pt! Terminar sessão'. The main heading is 'Recomendação' with the subtitle 'Informação do jogador'. The form contains two input fields: 'Email' and 'Período de Análise' (Analysis Period), which is pre-filled with '24/09/2019 - 08/10/2019'. Below the fields is a blue 'Gerar' (Generate) button and a blue 'Voltar' (Back) link.

Figura 37 - Obter recomendação



The screenshot shows the 'Criar Treino' (Create Training) page in the SmartCoach application. The page header includes 'SmartCoach' and user information 'Olá, admin@admin.pt! Terminar sessão'. The main heading is 'Criar Treino'. The form contains several input fields: 'Nome' (Name), 'Data' (Date) with a placeholder 'dd/mm/aaaa, --:--', and 'Duração (Min)' (Duration in Minutes). There is also a text area for 'Observações' (Observations). Below the fields is a 'Gerar Exercícios' (Generate Exercises) button with a gear icon. Underneath, there is a table with columns 'Nome', 'Duração (Min)', and 'Opções'. At the bottom of the form is a blue 'Criar' (Create) button and a blue 'Voltar' (Back) link.

Figura 38 - Criar treino com recomendação

De seguida, o utilizador preenche os dados para a criação do treino e seleciona a opção de gerar exercícios. Para a geração de exercícios com base na recomendação, é invocado um método denominado "Recommendation", que vai ser detalhado na **secção**

5.3, que tem em conta a recomendação recebida e a duração do treino que o utilizador pretende.

Após essa invocação, é gerado os exercícios e apresentado a informação, como é demonstrado na figura 38.

SmartCoach Olá, admin@admin.pt! Terminar sessão

Criar Treino

Nome

Data

Duração (Min)

Observações

Gerar Exercícios

Nome	Duração (Min)	Opções
Padronizado 6x0	20	Apagar
2x2+1 com cruzamento e finalização	20	Apagar
9x0	20	Apagar

[Criar](#) [Voltar](#)

Figura 39 - Treino com exercícios recomendados

5.2.7 Editar/Eliminar/Visualizar detalhes do treino

Esta funcionalidade traz ao utilizador a capacidade de editar, eliminar e visualizar os detalhes do treino.

Este caso de uso contém três possíveis ações do utilizador em relação aos treinos, editar treino, eliminar e visualizar os detalhes do mesmo.

Para o desenvolvimento do editar treino, a lógica é em tudo idêntica ao que foi abordado no **caso de uso 5.2.5**, sendo que difere na medida em que o *controller* inicialmente irá fazer um pedido ao repositório dos treinos para obter o treino pretendido através do ID, e assim permite ao utilizador obter todos os dados referentes ao mesmo e editá-los.

Na elaboração do caso de uso de visualizar detalhes, a *view* irá solicitar a informação ao *controller*, que inicialmente irá ter o mesmo comportamento explicado em cima, ou seja, irá efetuar um pedido ao repositório para obter o treino consoante o ID passado por parâmetro e assim fornecerá a informação.

A *view* irá apresentar a informação fornecida pelo *controller*, dando especial ênfase ao modo em como permite ao utilizador visualizar detalhes dos exercícios que estão inseridos no treino, ou seja, o utilizador quando está a visualizar os detalhes do treino tem a possibilidade de consultar qualquer exercício que esteja associado ao mesmo, como é demonstrado no código 3.

```
<div class="form-group">
  <label class="control-label">Exercícios: </label>
  <span class="text-danger" id="table_span" style="display: block;"></span>
  <div>
    <table class="table" id="tableExercicios">
      <thead>
        <tr>
          <th>
            <span>Nome</span>
          </th>
          <th>
            <span>Duração (Min)</span>
          </th>
        </tr>
      </thead>
      <tbody id="tbody1">
        @foreach (var exer in Model.Treino.Exercises)
        {
          <tr id="@exer.ExercicioId" class="exercicioTreino">
            <td>
              <a target="_blank" asp-page='/Exercicios/Details' asp-route-id="@exer.ExercicioId">
                @Html.DisplayFor(modelItem => exer.Exercicio.Name)
              </a>
            </td>
            <td>
              @Html.DisplayFor(modelItem => exer.Exercicio.Duracao)
            </td>
          </tr>
        }
      </tbody>
    </table>
  </div>
</div>
```

Código 3 - Exemplo View Detalhes do treino

A figura 39 é o exemplo representativo do *layout* da aplicação no visualizar detalhes do treino, onde é possível verificar a possibilidade de o utilizador seleccionar os exercícios que estão a azul.

SmartCoach Olá, admin@admin.pt! Terminar sessão

Detalhes

Treino

Nome Treino 27/09
Data 9/27/2019 9:00:00 PM
Duração (Min) 55
Observações

Exercícios:

Nome	Duração (Min)
9v9	20
Finalização 2x2+8J	15
Jogo reduzido 6x6+4J	20

[Editar](#) | [Voltar](#)

Figura 40 - Visualizar detalhes do treino

Por fim, o caso de uso de eliminar é idêntico ao explicado anteriormente, diferenciando-se por conter a opção de eliminar o treino. Quando o utilizador seleciona a opção de eliminar exercício, a view irá efetuar um pedido ao controller, e este consequentemente irá comunicar com o repositório de treinos de maneira a remover o mesmo da base de dados.

5.2.8 Visualizar calendário com agendamento dos treinos

A aplicação fornece, caso pretendido, ao utilizador a visualização do calendário com os agendamentos dos treinos.

Para a concretização deste caso de uso, foi desenvolvida a *View* através da utilização de plugin denominado de fullcalendar, que permitiu a criação do *layout* representado na figura 40.

O sistema através do controller comunica com um serviço, que posteriormente comunica com a base de dados e obtém todos os treinos criados pelo utilizador que tem a sessão iniciada na aplicação, e assim disponibiliza essa informação no calendário, como demonstra a figura seguinte.

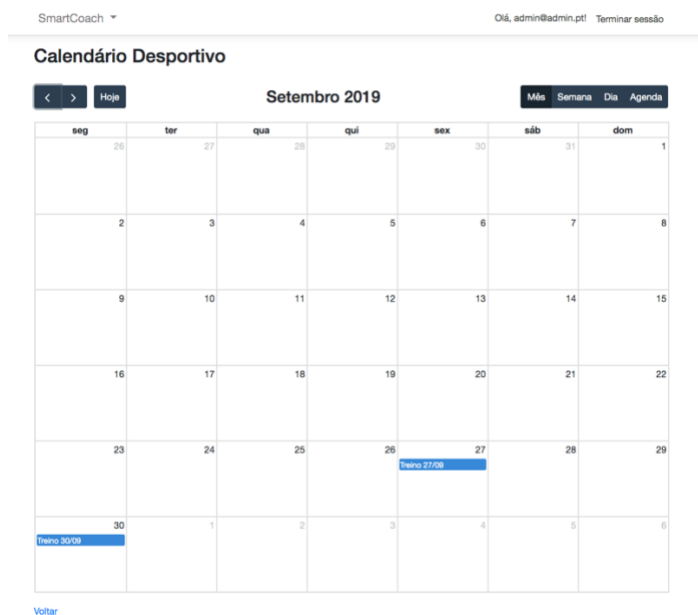


Figura 41 - Calendário desportivo

5.2.9 Dashboard estatístico

Utilizador tem a possibilidade de observar os dados estatísticos referentes aos atributos, exercícios e treinos.

No desenvolvimento deste caso de uso foi utilizado o plugin Highcharts, para a criação de todos os gráficos estatísticos necessários, auxiliado pelo *Model Dashboard*, onde estão presentes todas as informações que permitem gerar os gráficos relativos, ao número de treinos criados por mês do utilizador, exercícios e atributos.

Na Figura 41, está representado o layout da aplicação para este caso de uso.

Analisando a respetiva *View*, foram gerados dois gráficos de barras, em que é possível observar o crescimento de treinos criados pelo utilizador durante um ano, assim como, o número de atributos, exercícios e treinos que existe na base de dados.

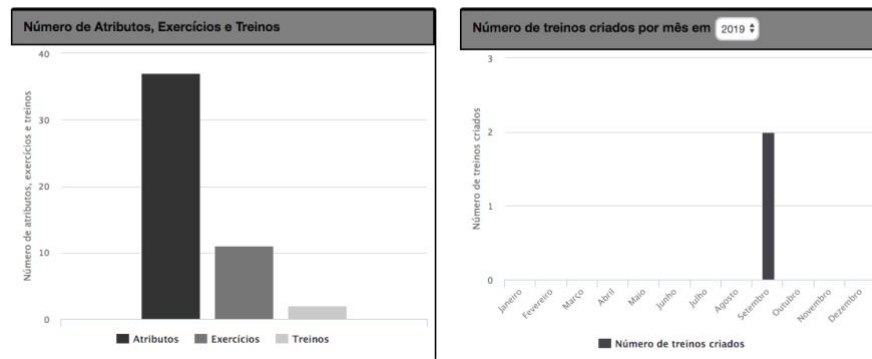
Dashboard Estatístico

Figura 42 - Dashboard Estatístico

5.3 Sistema de Classificação e Recomendação

Após a análise na secção anterior denominada de Casos de Uso, serão abordados os seguintes sistemas: o sistema de classificação de exercícios com base nos atributos associados e o sistema de recomendação de treinos. Irá ser explicado, de forma mais concreta, com o auxílio de excertos de código, todo o sistema e a sua respetiva comunicação com um sistema de recomendação externo.

A grande vantagem desta aplicação encontra-se na capacidade de classificar os atributos quanto ao exercício a que estão associados, sendo que, não existia nenhum software que preenchesse essa funcionalidade.

Inicialmente foram implementados os *models* necessários para a criação do exercício, e verificou-se a existência de uma relação de N para N entre o *model* ExercícioTreino (código 4) e Atributo, visto que um exercício pode conter vários atributos e um atributo pode estar em vários exercícios. Devido a esta relação e para não existir redundância de dados na base de dados, foi criado um model denominado AtributoAssignment como é demonstrado no código 5, que vai funcionar como um modelo intermediário que irá relacionar os modelos Atributo e ExercícioTreino.

Este modelo, conterà as referências para os dois modelos, que em termos de base dados, vai funcionar como chave estrangeira, e irá conter também o atributo classificação, garantindo assim que a classificação dada pelo utilizador é única, ou seja, só aquele exercício e especificamente o atributo é que tem a classificação introduzida.

```
public class ExercicioTreino
{
    public int Id { set; get; }

    [Display(Name = "Nome")]
    [Required, StringLength(100)]
    public string Name { set; get; }

    [Display(Name = "Descrição")]
    public string Description { set; get; }

    [Display(Name = "Duração (Min)")]
    public int Duracao { set; get; }

    [Display(Name = "Imagens")]
    public ICollection<FicheiroExercicio> Ficheiros { set; get; }

    [Display(Name = "Vídeo (Url)")]
    public string Url { get; set; }

    [Display(Name = "Atributos")]
    public ICollection<AtributoAssignment> AtributoAssignments { get; set; }

    public ICollection<ExercicioAssignment> Treinos { get; set; }

    public string UserId { get; set; }
}
```

Código 4 - Model ExercicioTreino

```
public class AtributoAssignment
{
    public int ExercicioID { get; set; }

    public int AtributoID { get; set; }

    public ExercicioTreino Exercicio { get; set; }

    public Atributo Atributo { get; set; }

    public double Classificacao { get; set; }
}
```

Código 5 - Model AtributoAssignment

Após a criação dos modelos, foi desenvolvido a *view* e o respetivo controller para a classificação dos atributos quanto ao exercício. Seguindo a explicação já abordada na **subsecção 5.2.3**, em que é possível analisar através do auxílio do diagrama de sequência toda a lógica implementada entre a *View*, *Controller*, *Repository* e os respetivos *Models*.

O sistema de recomendação está implementado numa api externa, denominada de SmartCoachApi.

Foi criado um *model* Recomendacao, como é demonstrado no código 6, para receber as informações relativas a uma recomendação através da api externa. Este modelo é constituído por informações como o nome, duração do treino e o utilizador associado á recomendação, visto que cada recomendação é única, e também por uma coleção de AtributoRecomendacao, ou seja, uma recomendação irá conter uma lista de atributos que é necessário melhorar.

```
public class Recomendacao
{
    public int ID { get; set; }

    public string Nome { get; set; }

    public int DuracaoTreino { get; set; }

    public ICollection<AtributoRecomendacao> AtributosRecomendacao { get; set; }

    public string UserId { get; set; }
}
```

Código 6 - Model Recomendacao

O *model* AtributoRecomendacao é constituído por referências ao atributo que deve ser melhorado e á respetiva recomendação e também pelo valor de recomendação, que é um valor compreendido entre zero e um (0 e 1), em que um (1) é o máximo valor de recomendação e o zero (0) o mínimo. O código 7 demonstra o *model* em questão.

```
public class AtributoRecomendacao
{
    public int ID { get; set; }

    public int AtributoID { get; set; }

    public Atributo Atributo { get; set; }

    public double ValorRecomendacao { get; set; }

    public int RecomendacaoID { get; set; }

    public Recomendacao Recomendacao { get; set; }
}
```

Código 7 - Model AtributoRecomendacao

Para ser fornecida a recomendação desejada, irá ser efetuado, através de um serviço *Gateway*, a comunicação de forma assíncrona com a api externa SmartCoachApi, como está presente na demonstração seguinte. O utilizador fornecerá os dados necessários, tais como, o email do jogador e o respetivo período de análise e a recomendação será obtida.

```

public static async Task<Recomendacao> GetRecomendacao(dynamic data)
{
    Recomendacao rec = null;
    HttpClient client = new HttpClient();
    client.BaseAddress = new Uri("https://smartcoachapi.azurewebsites.net/");
    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

    string dataJSON = JsonConvert.SerializeObject(data);
    HttpContent content = new StringContent(dataJSON, System.Text.Encoding.Unicode, "application/json");
    var response = await client.PostAsync("api/Treinos/GerarRecomendacaoJogadorBySCM", content);
    if (response.IsSuccessStatusCode)
    {
        string jsonString = await response.Content.ReadAsStringAsync();
        rec = JsonConvert.DeserializeObject<Recomendacao>(jsonString);
    }
    return rec;
}

```

Código 8 – Serviço de comunicação com api externa

Para a criação dos treinos, foi desenvolvido um *model* denominado Treino (código 9), que é constituído pelos seguintes elementos: nome, data do treino, duração, observações e uma coleção ExercícioAssignment.

```

public class Treino
{
    public int Id { get; set; }

    [Display(Name = "Nome")]
    public string Name { get; set; }

    [Display(Name = "Data")]
    public DateTime Date { get; set; }

    [Display(Name = "Duração (Min)")]
    public int Duration { get; set; }

    [Display(Name = "Observações")]
    public string Observations { get; set; }

    [Display(Name = "Exercícios")]
    public ICollection<ExercicioAssignment> Exercises { get; set; }

    public String UserId { get; set; }
}

```

Código 9 - Model Treino

Como foi explicado anteriormente, entre os *models* Treino e ExercicioTreino, existe uma relação de N para N, assim sendo, foi criado uma classe intermediária denominada de ExercicioAssignment, que está representada no código 10, e que é constituída pelas referências aos dois models em questão.

```
public class ExercicioAssignment
{
    public int ExercicioId { get; set; }

    public ExercicioTreino Exercicio { get; set; }

    public int TreinoId { get; set; }

    public Treino Treino { get; set; }
}
```

Código 10 - Model ExercicioAssignment

Após obter a recomendação necessária, o utilizador preenche os campos obrigatórios com os dados requeridos. Para os exercícios serem gerados, o algoritmo de recomendação desenvolvido necessita da recomendação e do tempo de duração definido pelo utilizador.

No código seguinte, é apresentado o algoritmo desenvolvido para gerar exercícios.

O algoritmo recebe por parâmetro a recomendação. Assim vai obter todos os atributos que devem ser melhorados e coloca-os numa lista de atributos auxiliar. Em seguida, comunicará com o repositório de exercícios, para obter todos os exercícios que contenham os atributos que devem ser melhorados.

Este algoritmo irá percorrer a lista dos exercícios e para cada atributo da Recomendação verifica se o exercício contém esse atributo, se sim, aplica a fórmula de cálculo apresentada na **Fórmula 1** para conseguir obter o valor de recomendação do exercício consoante a classificação do atributo em relação ao exercício associado e o valor de recomendação do atributo associado á Recomendação.

A fórmula 1 é uma soma ponderada dos valores obtidos de recomendação do exercício, ou seja, a cada atributo associado ao exercício é obtida a respetiva classificação e esta dividida por cinco (5) (valor máximo da classificação composta por estrelas), e de seguida é multiplicado pelo respetivo valor de recomendação oriundo da Recomendação e, a este resultado, é feito a soma ponderada do mesmo.

Com esta fórmula é possível calcular o valor de recomendação do exercício consoante os atributos presentes na recomendação obtida pelo sistema externo.

$$\text{valorExercicio} = \frac{\text{atributoClassificação}}{5} \times \text{valorRecomendação} + \text{valorExercicio}$$

Fórmula 1 - Cálculo do valor de recomendação do exercício

O algoritmo em cada exercício verifica todos os atributos da recomendação e adiciona esse exercício e o respetivo valor obtido pelo cálculo a um *Dictionary*.

Por fim, o *Dictionary* é ordenado de forma descendente consoante o *value* (é o valor da recomendação do exercício), para obter assim através das *keys*, só os objetos de exercício já ordenados do mais recomendado para o menos recomendado.

O algoritmo retorna os exercícios consoante a duração estabelecida pelo utilizador aquando da criação do treino, ou seja, se o utilizador estabeleceu sessenta (60) minutos a duração do treino, o algoritmo retorna os exercícios cujo a soma da duração dos mesmos, não ultrapasse a duração do treino.

```

public static class SistemaRecomendacao
{
    public static List<ExercicioTreino> Recommendation(ApplicationDbContext context, Recomendacao recomendacao)
    {
        Dictionary<ExercicioTreino, double> auxiliar = new Dictionary<ExercicioTreino, double>();

        List<Atributo> atributoTreinos = recomendacao.AtributosRecomendacao.Select(x => x.Atributo).ToList();

        List<ExercicioTreino> listaExercicios = context.Exercicios.Include(x => x.AtributoAssignments).ThenInclude(y => y.Atributo).
        Where(x => x.AtributoAssignments.Any(z => atributoTreinos.Any(y => y.Name == z.Atributo.Name))).ToList();

        foreach (var exercicio in listaExercicios)
        {
            double res = 0;

            foreach (var rec in recomendacao.AtributosRecomendacao)
            {
                foreach (var atr in exercicio.AtributoAssignments.Where(x => x.Atributo.Name == rec.Atributo.Name))
                {
                    res = (atr.Classificacao / 5) * rec.ValorRecomendacao + res;
                }
            }

            auxiliar.Add(exercicio, res);
        }

        var lista = auxiliar.OrderByDescending(x => x.Value).ToDictionary(x => x.Key, x => x.Value).Keys.ToList();
        var result = new List<ExercicioTreino>();
        var count = 0;

        foreach (var exe in lista)
        {
            if (count + exe.Duracao <= recomendacao.DuracaoTreino)
            {
                result.Add(exe);
                count+=exe.Duracao;
            }
        }

        return result;
    }
}

```

Código 11 - Algoritmo de Recomendação

6 Avaliação

Neste capítulo é exposto como se realizará o processo de avaliação para as principais características a avaliar do projeto. Assim sendo, primeiramente são identificadas as métricas e grandezas que devem ser avaliadas em relação ao projeto. De seguida, é apresentada e justificada a metodologia a utilizar na avaliação e são demonstrados os respetivos testes e análises.

6.1 Grandezas

Considerando o tipo de projeto em questão e o conjunto de funcionalidades que a solução idealizada propõe, as grandezas a considerar prendem-se com a exatidão dos cálculos realizados no módulo de análise de rentabilidade de projetos e a satisfação do utilizador em duas vertentes: utilização geral da aplicação e satisfação obtida consoante a utilização do módulo de classificação e em relação aos resultados apresentados pelo mesmo.

Serão então utilizadas duas grandezas diferentes para a avaliação de todo o projeto:

- **Avaliar internamente o sistema:** É sempre importante avaliar as especificidades de um produto, principalmente, quando se trata de um software de apoio à gestão desportiva, desse modo, deverá ser analisada e avaliada a exatidão e o

tempo dos valores proveniente dos algoritmos de classificação, assim como de outras funcionalidades relevantes do sistema.

- **Avaliar a satisfação do utilizador:** Noutra vertente, a satisfação do utilizador para o projeto em questão, é, provavelmente o fator mais importante a ter em consideração, isto considerando que esta é uma aplicação de suporte para múltiplos utilizadores, e é a opinião dos mesmos que irá condicionar o futuro deste projeto.

6.2 Metodologia

Tendo em vista testar e avaliar o sistema consoante as grandezas estabelecidas anteriormente, foram identificadas duas metodologias a pôr em prática. Sendo estas de seguida apresentadas ambas as metodologias:

Na sequência da grandeza **“Avaliar internamente o sistema”**, podemos concluir que o sistema no geral deveria ser testado e avaliado, para isso deverá ser adotado modelo V, que se rege sobre as normas de verificação e validação do software ao longo do seu desenvolvimento. No entanto, no sentido de averiguar a exatidão dos resultados obtidos pelo sistema de classificação, o mesmo será verificado através de testes unitários, integração e sistema.

Já no caso de apurar o tempo despendido nos algoritmos de classificação, o sistema terá de registar os vários tempos gastos, desde a chamada até ao resultado da classificação alcançada, possibilitando assim, a realização de uma média de tempos obtidos para concluir assim a sua avaliação.

Já em relação à grandeza **“Avaliar a satisfação do utilizador”**, primeiro, será disponibilizado o software produzido para duas equipas juvenis de futebol, para as mesmas puderem testar o produto num caso real. Após um mês de utilização, será obtida a opinião dos utilizadores através de um inquérito relacionado com a satisfação

do utilizador, este tem como objetivo avaliar a usabilidade, a eficiência, e claro a utilidade da aplicação desenvolvida.

Mais especificamente, o inquérito partilhado deverá dispor de um vasto número de questões tendo em consideração o papel do utilizador na aplicação e das funcionalidades existentes.

Para cada questão será apresentada uma escala de possíveis respostas de um até cinco (1 a 5), que vão desde a denominação de “muito insatisfeito”, até “muito satisfeito”. Após estas questões, irá existir uma final opcional, na qual, o utilizador tem a possibilidade de expor a sua opinião sobre o respetivo software, assim como, possíveis melhorias que gostaria de ver implementadas no futuro.

6.2.1 Modelo V

Como é referido na grandeza de avaliar o sistema internamente, foi adotado o modelo V para avaliar a aplicação. O modelo V exige que cada produto resultante da atividade de desenvolvimento de software seja verificado e validado, permitindo assim a identificação de problemas o mais cedo possível.

O teste pretende assegurar que as especificações do sistema se encontram corretamente implementadas, e que, a solução vai de encontro aos requisitos de negócio e aos requisitos técnicos. Com a adoção deste modelo espera-se uma redução substancial do número de erros na fase final do projeto.

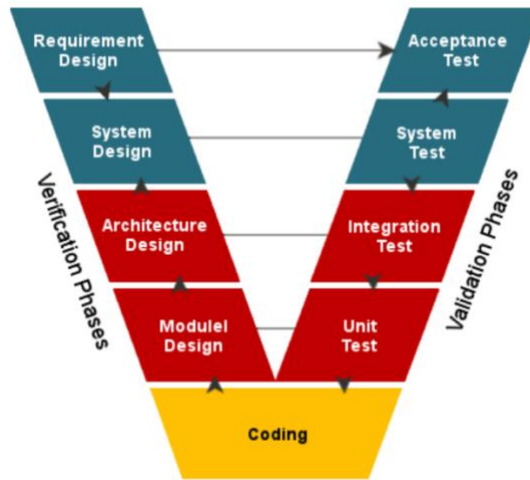


Figura 43 - Modelo V³³

A figura 42, pretende dar uma visão do modelo em questão.

6.3 Implementação dos testes

Nos objetivos iniciais do projeto foi definido que iriam ser implementados testes unitários, integração e sistema. A figura 43, demonstra a organização do projeto SmartCoachManager para a estrutura onde será desenvolvido os testes.

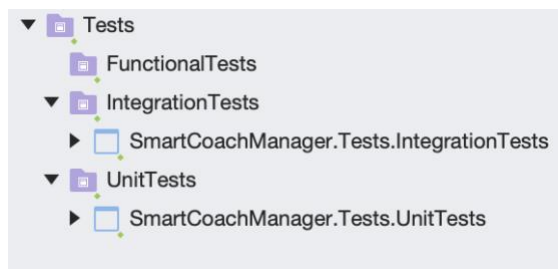


Figura 44 - Organização dos testes da aplicação "SmartCoachManager"

Nas seguintes secções serão abordados a implementação de cada um destes testes.

³³ Sonali Mathur, Shaily Malik; Advancements in the V-Model

6.3.1 Testes Unitários

Os testes unitários não têm como objetivo testar como o código funciona com outras dependências externas, mas sim, testar de forma individual, a lógica aplicacional.

Estes, caracterizam-se por serem testes com tempos de execução baixos, e por essa razão devem ser executados com bastante frequência.

Neste projeto os testes unitários foram desenvolvidos sobre as entidades de domínio e os serviços aplicacionais.

No código 12, está representado um exemplo de teste unitário implementado no projeto, onde é possível verificar que foi seguido a nomenclatura dos casos de teste, assim, é possível, pelo nome perceber o teste que vai ser executado, neste caso ao cálculo do valor da recomendação.

```
[Fact]
public void calculoValorRecomendacaoTest()
{
    var atributoClassificacao = 2;

    var res = 0;

    var valorAtributoRecomendacao = 0.80;

    var resultadoEsperado = 0.32;

    var resultadoObtido = SistemaRecomendacao.calculoValorRecomendacao(res, atributoClassificacao, valorAtributoRecomendacao);

    Assert.True(resultadoObtido == resultadoEsperado);
}
```

Código 12 - Teste Unitário

6.3.2 Testes de Integração

Os testes de integração têm como objetivo expor as falhas das interações entre os diferentes componentes do sistema. Este género de teste procura validar o sistema como um todo.

Foi realizado este género de testes através da *framework* postman, com o auxílio desta ferramenta, é possível testar os serviços desenvolvidos na api; e foi através desta *Framework* foi criada uma coleção de testes.

Na figura seguinte, é demonstrado um dos testes de integração realizados ao sistema, neste caso específico, é testada a comunicação com a api externa quando é efetuado o pedido para obter uma recomendação para o treino.

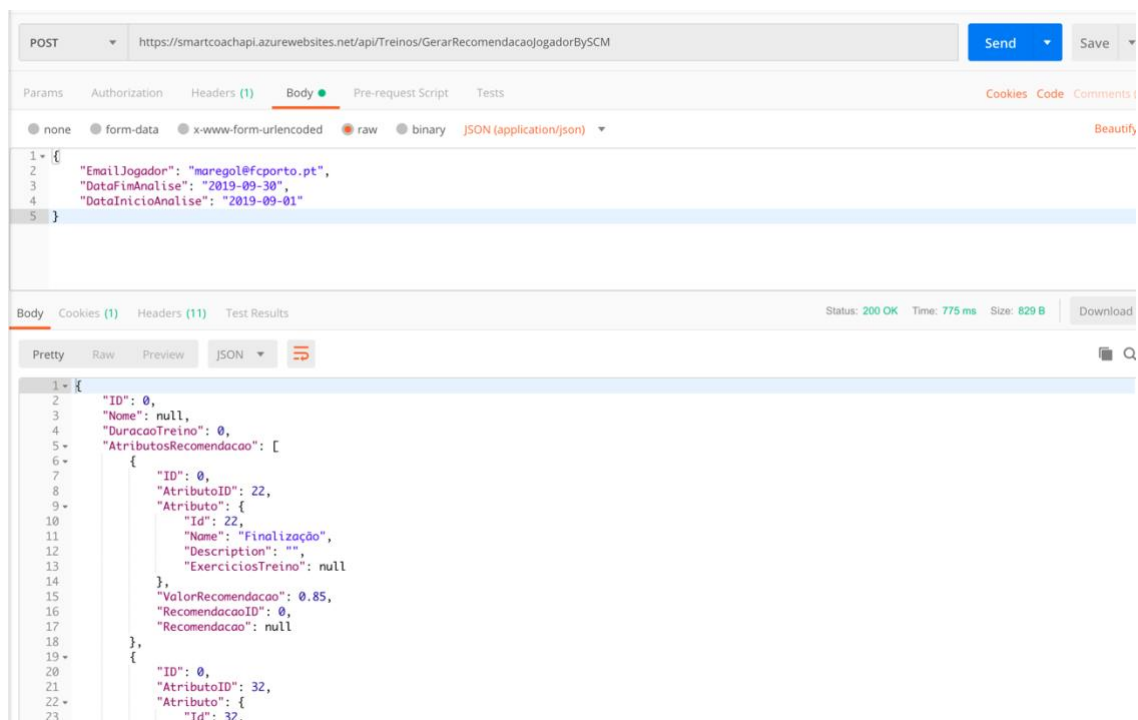


Figura 45 - Teste Integração GetRecomendation

Foram efetuados este tipo de testes sempre que ocorria um pedido a um serviço da api, assim, foi possível testar todos os serviços da aplicação.

6.3.3 Testes de Sistema

Os testes de sistema determinam se o software contém todos os requisitos definidos na especificação de requisitos de software. A execução destes é demonstrada no formato de tabela, onde é descrita a ação realizada, o resultado esperado e o resultado final.

Na tabela 9, está representado um teste de sistema efetuado à aplicação, neste exemplo foi testada a funcionalidade de *login* na aplicação e consequente visualização do calendário desportivo.

Tabela 9 - Teste de sistema

Teste de Sistema Nº1	
Descrição:	Utilizador realiza o início da sessão na aplicação, e seleciona a opção de visualizar calendário desportivo.
Resultado Esperado:	O <i>login</i> é realizado com sucesso, e de seguida é apresentado o calendário de agendamento dos treinos.
Resultado Obtido:	PASSOU

Foi realizado também um teste de carga em torno do tempo de resposta do algoritmo de Recomendação de treinos. No seguinte gráfico, podemos visualizar as oscilações de valor relativas ao tempo de resposta a cem (100) chamadas ao método.

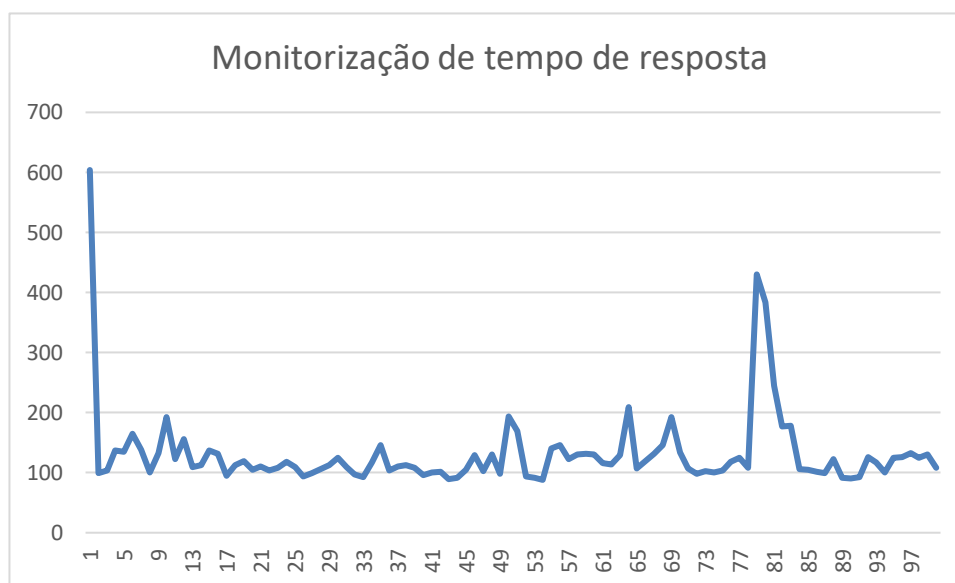


Figura 46 – Gráfico monitorização do tempo de resposta do algoritmo

Através do gráfico anterior, foi realizada uma tabela onde está organizada toda a informação essencial relativa ao mesmo. Concluindo então, sendo o valor da Média de resposta do algoritmo cento e trinta e um (131) milésimos de segundo, que o tempo de resposta a este é bastante positivo.

Tabela 10 - Valores referentes á monitorização

Média	Desvio Padrão	Máximo	Mínimo
131,38 ms	68,337	604	88

6.3.4 Testes de aceitação

Os testes de aceitação têm como propósito validar a aplicação em relação aos requisitos do cliente. A execução destes testes envolve habitualmente a interação de um utilizador com a aplicação, por exemplo, através de ações ou validações sobre a *interface*.

Estes, foram realizados através de uma versão Beta do protótipo disponibilizada a vários potenciais clientes finais, e com o auxílio de questionários foi recolhida a sua opinião em relação á mesma. A secção **6.4.2** apresentará uma análise mais específica em relação aos questionários em questão.

Na tabela 11, está representado um exemplo de um teste de aceitação realizado à aplicação, este caso concreto, à funcionalidade de iniciar sessão na aplicação com sucesso.

Tabela 11 - Teste de aceitação

Teste de Aceitação Nº1	
Descrição:	Iniciar a sessão na aplicação com sucesso.
Resultado Esperado:	O utilizador não autenticado inicia sessão na aplicação.
Resultado Obtido:	PASSOU

6.4 Avaliação de usabilidade

Esta secção é dedicada à avaliação de usabilidade e esta é realizada à aplicação desenvolvida. Para efetuar esta avaliação, foi realizada uma sessão de demonstração do funcionamento da aplicação a um diretor da escola de futebol Hernâni Gonçalves, e também, a alguns membros do GECAD.

Posteriormente foi disponibilizado um guião de testes, acompanhado de um questionário online, para os participantes responderem. O principal objetivo deste, era recolher a opinião do utilizador sobre o funcionamento e aspeto visual das aplicações “SmartCoach” e “SmartCoachManager”.

Depois da recolha dos questionários, irá ser realizada uma análise para se obter as conclusões das opiniões da amostra recolhida, contudo, apenas serão abordadas as opiniões relativas á aplicação “SmartCoachManager”.

6.4.1 Metodologia para a sessão de avaliação

Primordialmente, foi realizada uma demonstração da aplicação a um diretor da Escola de Futebol Hernâni Gonçalves e alguns elementos do GECAD. Esta demonstração, foi realizada presencialmente no dia 03/10/19 no ISEP, tendo uma duração de aproximadamente uma hora e trinta minutos. No final da sessão, foi disponibilizado via e-mail um guião de testes acompanhado por um questionário online.

O guião para a avaliação de usabilidade é constituído pelos seguintes elementos: Introdução, onde é referido o seu objetivo e as pré-configurações dos ambientes de testes, e o cenário de teste em avaliação. Este, pretende analisar o seguinte caso:

- Utilizador não registado- Pretende simular-se o momento em que o Utilizador tem o primeiro contacto com a aplicação.

O questionário final foi realizado no *Google Forms*, uma plataforma online disponibilizada pela Google, que permite a criação e desenvolvimento de questionários, esta, com as respostas aos mesmo, consegue criar um ficheiro Excel com a informação

devidamente organizada que permite, por exemplo, a criação de gráficos e tabelas com os dados recolhidos.

Todas as questões do guião seguem a escala de *Likert*. O principal objetivo desta, é medir o nível de concordância ou não, referente a uma afirmação. Esta escala, descreve um certo atributo num intervalo de valores e, cada valor da escala, corresponde a um nível diferente desse atributo. Na figura 47, está representado a escala utilizada no questionário de usabilidade.

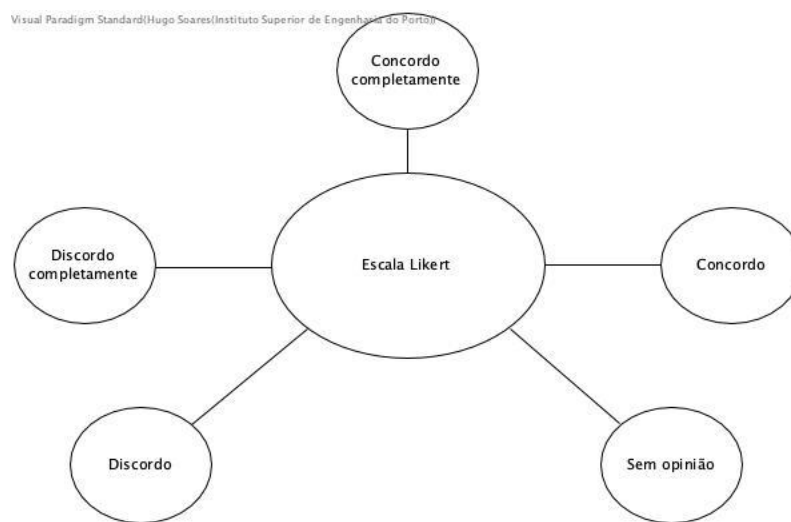


Figura 47 - Escala Likert

Apesar de se considerar que uma amostra de dez inquiridos não é significativa, os estudos de Virzi pretendem desvalorizar esta afirmação mostrando que uma amostra populacional de cinco inquiridos, consegue identificar aproximadamente oitenta por cento (80%) dos problemas de usabilidade. Assim sendo, consoante esta teoria, e sendo que a amostra é constituída por dez inquiridos, espera detetar-se cerca de noventa por cento (90%) dos problemas de usabilidade.

6.4.2 Resultados e interpretação de resultados

Esta subsecção demonstrará todos os resultados obtidos que provêm dos questionários realizados e a sua respetiva análise.

A amostra é constituída por dez indivíduos, dos quais cinco são jogadores, dois treinadores, dois elementos do staff técnico e um utilizador associado (indivíduo externo ao clube, ex.: pai de um atleta). Esta análise é respetiva á figura 48- Cargos referentes à amostra.

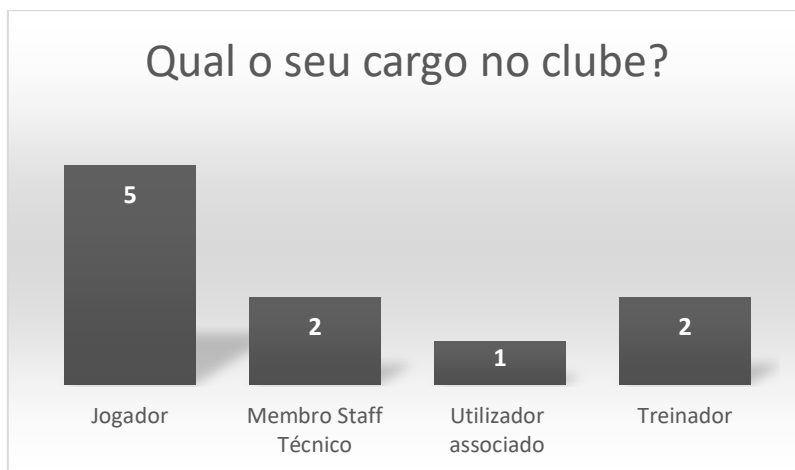


Figura 48 - Cargos referentes à amostra

O gráfico ilustrado pela Figura 49, representa a opinião dos inquiridos relativamente a questões relacionadas com algumas características da aplicação tais como: design, tempo de resposta e número de funcionalidades. Primeiramente, importa explicar a legenda do gráfico, que por uma questão de *design* as perguntas ficaram com a legenda de “pergunta”. Assim sendo, na tabela 12 encontramos as legendas adequadas relativas as perguntas do questionário.

Tabela 12 – Legenda do gráfico “Design e Usabilidade”

Pergunta 1	A aplicação tem todas as funcionalidades desejadas?
Pergunta 2	Considera a aplicação simples e intuitiva?
Pergunta 3	A aplicação possui uma boa apresentação e design?
Pergunta 4	Considera a aplicação com tempo de resposta curto?

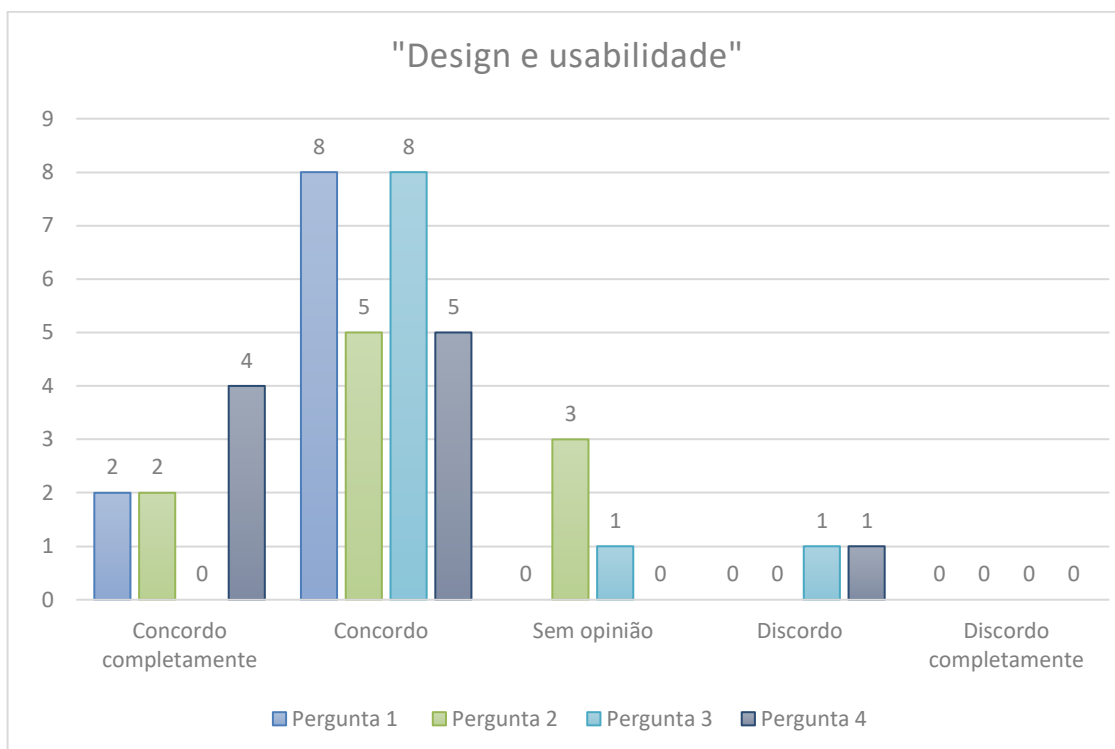


Figura 49 – Gráfico de opiniões “Design e usabilidade”

Entendendo que, as hipóteses “Concordo e Concordo Completamente” são as hipóteses mais favoráveis às questões, ao analisar o gráfico, podemos observar que a maior parte dos inquiridos na maioria das questões possui uma opinião favorável relativamente ao Design e a usabilidade da aplicação.

No gráfico seguinte será analisada a componente referente a esta dissertação, “SmartCoachManager”, em termos de funcionalidades e design. Na tabela 13 encontramos as legendas adequadas relativas as perguntas do questionário.

Tabela 13 - Legenda do gráfico "SmartCoachManager"

Pergunta 1	Criação de exercícios de treino tem todos os parâmetros pretendidos?
Pergunta 2	Considera a classificação do exercício com base em atributos uma mais valia?

Pergunta 3	Considera o processo de criação de um treino (manual ou com recomendação) simples e intuitivo?
Pergunta 4	Considera a calendarização dos planos de treino adequada?

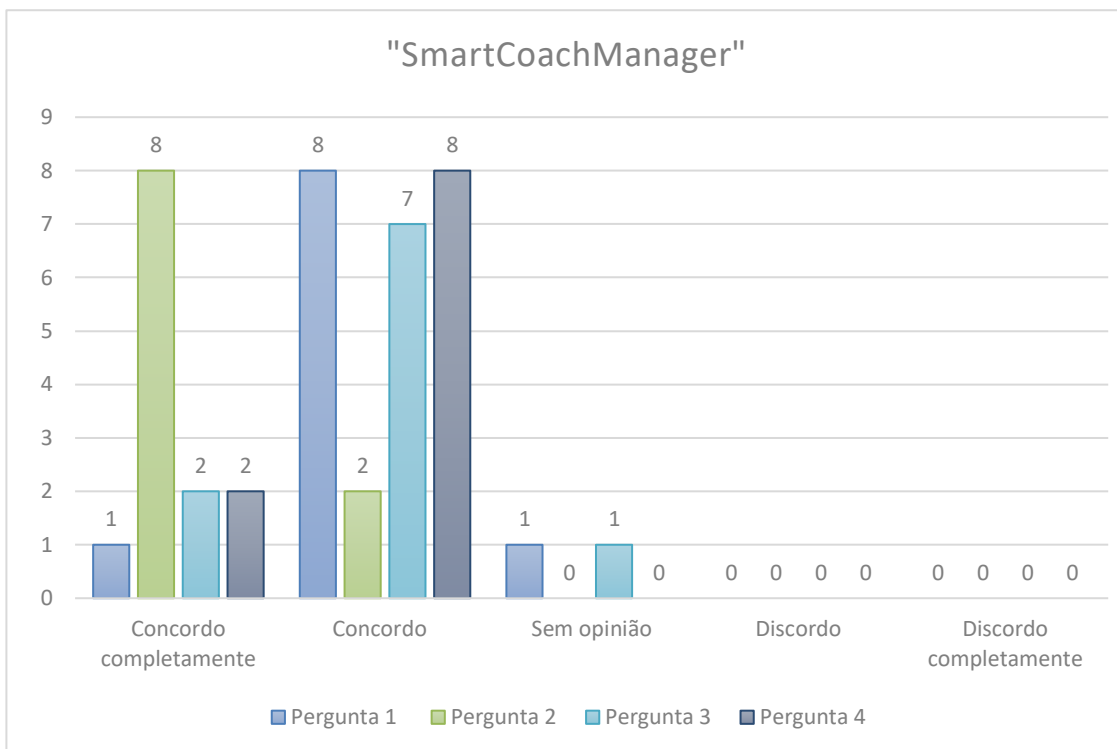


Figura 50 - Gráfico de opiniões "SmartCoachManager"

Ao observarmos o gráfico de opiniões "SmartCoachManager" da Figura 48, observamos que na Pergunta 1 e 2, cem por cento (100%) dos inquiridos deram as respostas "Concordo completamente e Concordo", sendo que na pergunta 3 e 4, só observamos dez por cento (10%) da população inquiridas que não tem opinião sobre tal e o remanescente, noventa por cento (90%) tem uma opinião favorável relativamente às perguntas.

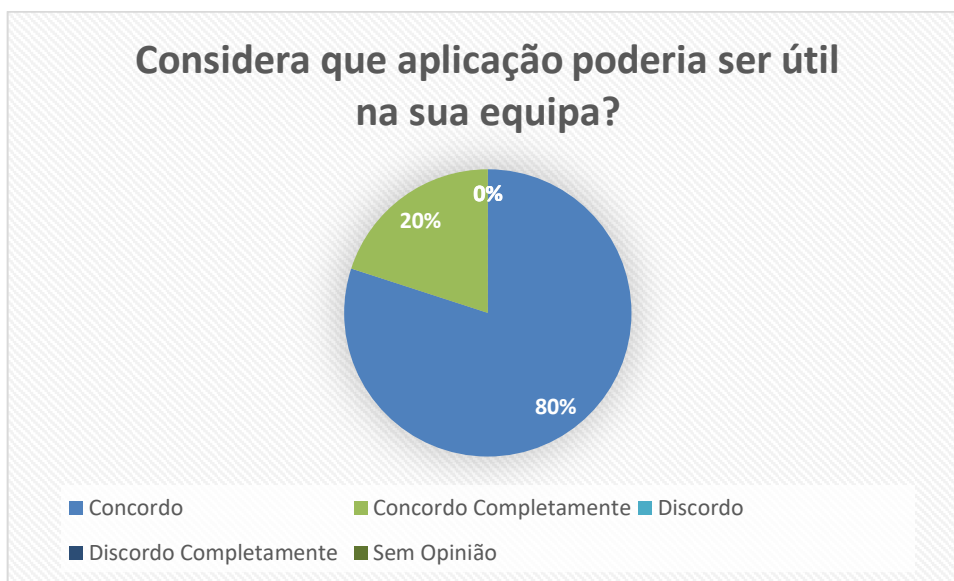


Figura 51 - Gráfico de opiniões Utilidade da aplicação

Neste gráfico, podemos verificar que todos inquiridos consideram a aplicação útil na sua equipa, sendo que cem por cento (100%) responderam às questões com “Concordo e Concordo Completamente”.

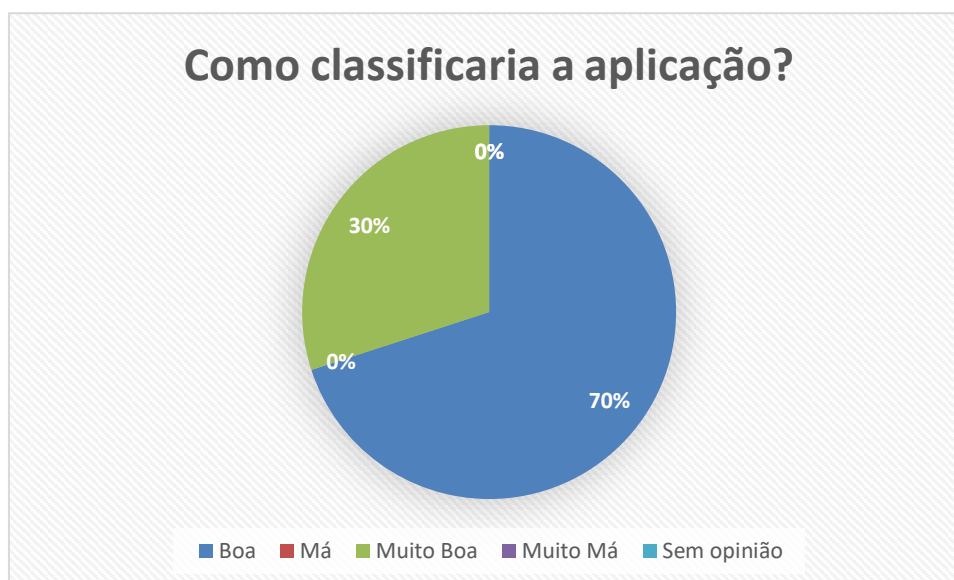


Figura 52 - Gráfico- Classificação da aplicação

Como podemos observar, toda a amostra tem boas considerações relativas à aplicação, sendo que, trinta por cento (30%) responderam “Muito Boa” e setenta por cento (70%) responderam “Boa” à seguinte questão “Como classificaria a aplicação”.



Figura 53 - Gráfico de opiniões Recomendação da aplicação

Este gráfico mostra que, mais da maioria dos inquiridos, noventa por cento (90%) recomendaria a aplicação a outros utilizadores, e apenas dez por cento dos inquiridos têm dúvidas se o fariam, como podemos concluir da resposta “Talvez”.

Na última questão os inquiridos podem exprimir sua opinião ou comunicar sugestões de melhoria relativamente á aplicação. Na tabela 14 podemos observar as sugestões dos inquiridos.

Tabela 14 - Sugestões de melhorias e comentários dos inquiridos

Design, exercícios mais variados, intuitividade
Melhor Design
Com a individualidade ser capaz de criar estatísticas coletiva, exercícios específicos para a equipa e não individualmente.

6.4.3 Apreciações finais

A avaliação da usabilidade da aplicação teve como objetivo analisar a interação dos participantes com a aplicação. A elaboração do guião foi essencial para o sucesso do questionário, já que, nem todos os inquiridos estiveram presentes na reunião de demonstração.

No que diz respeito aos gráficos relativos ao *design* e usabilidade da aplicação, as respostas foram positivas. Apesar dos gráficos apresentarem duas respostas “Discordo” e três “Sem opinião” não representam um número significativo para serem relevantes, até porque o remanescente das trinta e sete respostas apresentam resultados positivos como “Concordo Completamente e Concordo”.

Quanto ao gráfico relativo à componente SmartCoachManager, as respostas foram extremamente positivas, visto que no total das quarenta respostas obtidas, trinta e oito foram respostas como “Concordo Completamente e Concordo”, sendo que só duas respostas é que foram “Sem opinião”, o que permite concluir que em relação à componente de SmartCoachManager os utilizadores estão bastante satisfeitos com o design e as funcionalidades da aplicação.

No ímpeto geral, noventa por cento (90%) dos utilizadores recomendariam a aplicação a novos utilizadores, setenta por cento (70%) classificou a aplicação como boa e o restantes trinta por cento (30%) como muito boa, e por fim, a totalidade da população considera a aplicação útil para as respetivas equipas. Dado isto, a avaliação final dos questionários permitiu observar que as respostas relativas à aplicação são altamente positivas.

7 Conclusões

Neste capítulo são apresentadas as conclusões desta dissertação.

Na secção 7.1 é efetuada uma descrição sucinta do trabalho desenvolvido, na secção 7.2 é realçado os objetivos atingidos. As limitações da solução desenvolvida e as ideias de trabalho futuro estão descritas na secção 7.3.

7.1 Resumo

O presente projeto surgiu da intenção de melhorar a performance dos jogadores de uma equipa de futebol, auxiliando-os com um sistema de classificação de exercícios por atributos, permitindo assim ao utilizador criar um plano de treino focado nos atributos que deseja melhorar.

Embora existam inúmeras aplicações que, através de diferentes métodos e técnicas tentem minimizar este problema, verificou-se uma limitação comum à solução que consistia na classificação dos exercícios por atributo e assim possibilitar uma recomendação de um plano de treino, dado que, as aplicações existentes no mercado atual, são extremamente focadas na gestão de equipa, estatísticas e criação de exercícios de forma interativa.

No capítulo 1 foi realizada uma breve contextualização, a definição do problema, os objetivos estipulados e resultados esperados.

No capítulo 2 foi apresentado o estado da arte, onde se definiram vários conceitos essenciais para a dissertação, com estes foi possível concluir que não foi encontrado softwares no mercado com exatamente as mesmas características, o que torna este projeto inédito.

No capítulo 3 foi apresentada a análise de valor.

No capítulo 4 foi realizada a análise o *design* da solução a desenvolver. Foram apresentados: o modelo de domínio e os requisitos funcionais e não-funcionais identificados, requisitos esses que vieram a definir a arquitetura final da solução. Foi realizada a análise da melhor arquitetura a implementar. Por fim, foi apresentada uma possível solução com base no modelo de arquitetura 4 + 1.

No capítulo 5 foram apresentados os detalhes de implementação da solução desenvolvida, onde se procurou detalhar os aspectos mais significativos, tais como alguns detalhes de implementação da aplicação *web*, assim como de cada caso de uso implementado.

Posteriormente, no capítulo 6, foi discutida a avaliação da solução. Verificou-se que foram realizados testes unitários, de integração, de sistema e de usabilidade. Já na análise dos resultados dos testes de usabilidade, concluiu-se que, de forma geral, os participantes do questionário concordam que a plataforma é útil e que pode ser importante no dia-a-dia de um clube.

7.2 Objetivos Alcançados

Inicialmente foram definidas metas de desenvolvimento para o projeto, na tabela seguinte serão expostos os objetivos e os seus respectivos estados, que podem variar entre: completo, parcialmente completo, incompleto, não realizado.

Tabela 15 - Objetivos propostos e estado de conclusão

Objetivo	Estado
Levantamento e especificação de requisitos	Completo
Desenvolvimento da arquitetura	Completo
Desenvolvimento da aplicação	Completo
UI Responsiva	Completo
Sistema classificação de exercícios	Completo
Integração com sistema de recomendação	Completo
Realização de testes unitários	Completo
Realização de testes de integração	Parcialmente Completo
Realização de testes de sistema	Completo
Realização de testes de usabilidade	Completo

Podemos afirmar, através do auxílio da tabela 15, que os objetivos iniciais e que foram surgindo ao longo da realização do projeto foram todos alcançados com sucesso. A realização de testes de integração está parcialmente completa, visto que só foram realizados testes através do postman e não foram implementados na aplicação.

Ao longo da realização desta dissertação de mestrado, foram desenvolvidos alguns artigos científicos referente á mesma. Em seguida, serão expostas as informações relativas aos mesmos:

- Autores: Matos, Paulo and Rocha, João and Gonçalves, Ramiro and Almeida, Ana and Santos, Filipe and Abreu, David and Soares, Hugo and Martins, Constantino; Hybrid recommendation system for young football athletes customized training³⁴
- Autores: Matos, Paulo and Rocha, João and Gonçalves, Ramiro and Almeida, Ana and Santos, Filipe and Abreu, David and Soares, Hugo and Martins, Constantino; Information system for young football athletes customized training³⁵

7.3 Limitações e Trabalho Futuro

Ao longo do desenvolvimento do projeto foram identificadas algumas limitações que poderão ser alvo de futuras melhorias.

Uma das quais, é o projeto não recolher o feedback dos jogadores quanto aos treinos, para posteriormente esse mesmo feedback ser uma variável importante no algoritmo de recomendação, ou seja, se um jogador atribuir a classificação de “mau” ao treino recomendado, o sistema na recomendação seguinte terá em conta essa classificação, para não repetir o mesmo treino.

A aplicação quanto á realização de testes, deve ser implementado testes de integração.

Sendo este projeto uma versão demo, não foram desenvolvidos mecanismos de segurança no que diz respeito à política de proteção de dados (RGPD). Sendo esta uma política essencial na atualidade, seria interessante o desenvolvimento deste requisito.

³⁴ [Matos et al., 2020 in press] Matos, P., Rocha, J., Gonçalves, R., Almeida, A., Santos, F., Abreu, D., Soares, H., and Martins, C. (2020 in press). Hybrid recommendation system for young football athletes customized training. In *Future of Information and Communication Conference, Lecture Notes in Networks and Systems. Advances in Information and Communication*, Springer.

³⁵ [Matos et al., 2019] Matos, P., Rocha, J., Gonçalves, R., Almeida, A., Santos, F., Abreu, D., Soares, H., and Martins, C. (2019). Information system for young football athletes customized training. In *19a Conferência da Associação Portuguesa de Sistemas de Informação*

Referências

R. Jacinto and P. Malta. [Online]. Available: <http://hdl.handle.net/10316.2/40645>.

A. Couto, “easyApp,” 8 Agosto 2016. [Online]. Available: <https://www.easyapp.pt/blog/apps-de-saude-e-bem-estar/>.

J. Taira, “Visão,” 27 Novembro 2015. [Online]. Available: <http://visao.sapo.pt/opiniao/bolsa-de-especialistas/2015-11-27-A-importancia-da-tecnologia-no-futebol>. [Accessed Fevereiro 2019].

L. Rama, “idesporto,” 2016. [Online]. Available: http://www.idesporto.pt/ficheiros/file/Manuais/Graul/Graul-07a_Metodologia%20Individ.pdf.

L. Filipe, “Sistema de Recomendação Baseado em Conhecimento,” Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, 2012.

“tacticalsoccer,” [Online]. Available: <https://www.tacticalsoccer.pt>.

“coach-helper,” [Online]. Available: <https://www.coach-helper.com>.

“mycoachfootball,” [Online]. Available: <https://www.mycoachfootball.com/en/>.

“soccer-trainer,” [Online]. Available: <http://www.soccer-trainer.com.pt>.

“newsoccerdrills,” [Online]. Available: <https://www.newsoccerdrills.com/en/>.

PeterA.Koen, Fuzzy Front End: Effective Methods, Tools, and Techniques.

I. Rish, "An empirical study of the naive Bayes classifier," T.J. Watson Research Center.

J. QUINLAN, "Induction of Decision Trees," Kluwer Academic Publishers, Boston, 1986.

P. Cunningham and S. J. Delany, "k-Nearest Neighbour Classifiers".

J. Schmidhuber, "Neural Networks," 2015.

M. J. L. Orr, "Introduction to Radial Basis Function Networks," Edinburgh, 1996.

J. Lafferty, A. McCallum and F. C. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," University of Pennsylvania Scholarly Commons, Pennsylvania, 2001.

K. Papineni, "Why Inverse Document Frequency?," IBM T.J. NY.

M. J. Culnan, "The Intellectual Development of Management Information Systems, 1972–1982: A Co-Citation Analysis," 1986.

I. Dagan, L. Lee and F. C. N. Pereira, "Similarity-Based Models of Word Cooccurrence Probabilities," 1999.

[Matos et al., 2020 in press] Matos, P., Rocha, J., Gonçalves, R., Almeida, A., Santos, F., Abreu, D., Soares, H., and Martins, C. (2020 in press). Hybrid recommendation system for young football athletes customized training. In *Future of Information and Communication Conference*, Lecture Notes in Networks and Systems. Advances in Information and Communication, Springer.

[Matos et al., 2019] Matos, P., Rocha, J., Gonçalves, R., Almeida, A., Santos, F., Abreu, D., Soares, H., and Martins, C. (2019). Information system for young football athletes customized training. In *19ª Conferência da Associação Portuguesa de Sistemas de Informação*

Sonali Mathur, Shaily Malik; *Advancements in the V-Model*

Scott Millett, Nick Tune, *Patterns, Principles and Practices of Domain-Driven Design*

A. Leff, J. T. Rayfield, *Web-application development using Model/View/Controller design pattern*. 2001

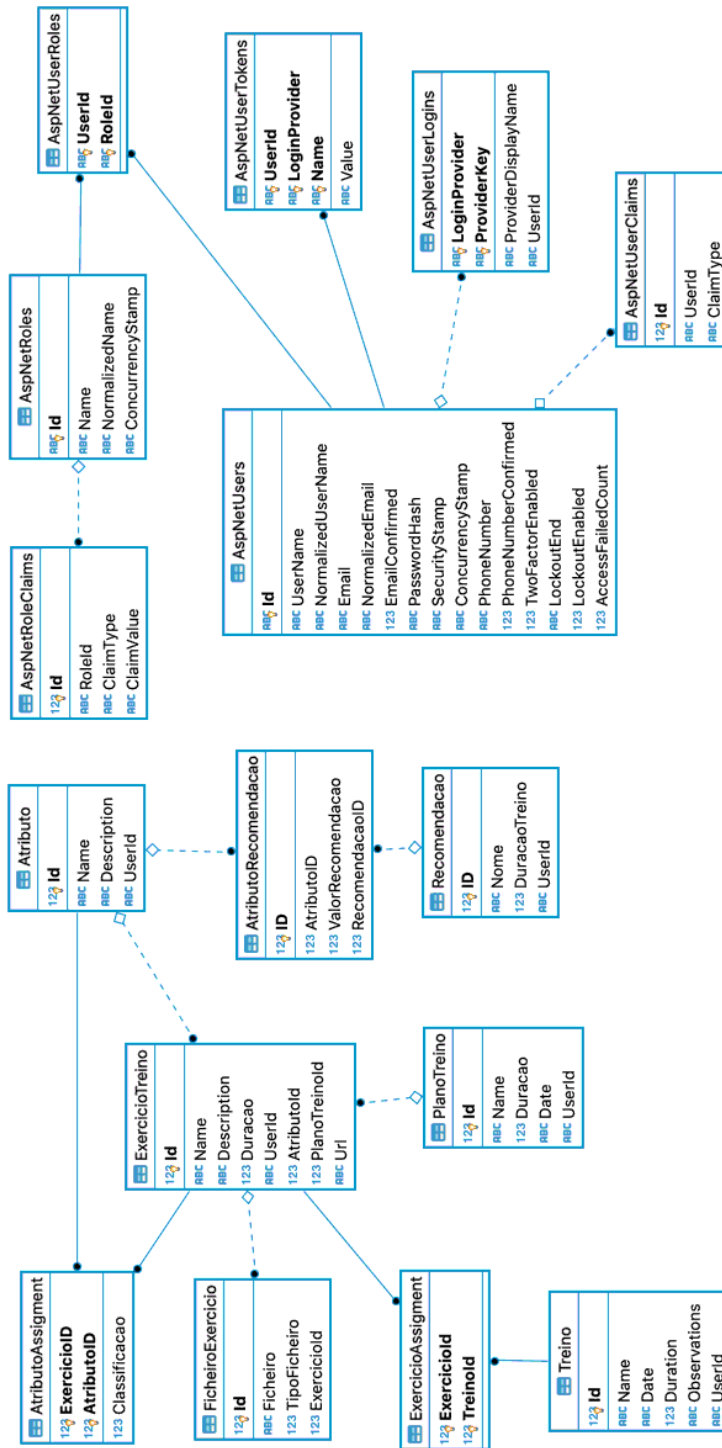
JOSUÉ LUCIANO; WALLISON JOEL BARBERÁ ALVES, *PADRÃO DE ARQUITETURA MVC: MODEL-VIEW- CONTROLLER*

P. B. Kruchten, *The 4+1 View Model of architecture*. 1995

M.Zilber, F.Bonazzi; Innovation and Business Model: a case study about Integration of Innovation Funnel and Business Model Canvas. 2014

Anexos

Anexo 1 – Modelo de dados ER



Anexo 2 – Guião de testes



Guião de Testes

Plataforma SmartCoachManager

Objetivo

Este guião tem o objetivo de orientar o processo de testes da aplicação SmartCoachManager. Este, serve como guia, para a amostra de utilizadores puder aceder á aplicação e compreender as suas funcionalidades e características. Assim, estes são capazes de responder, consoante a sua experiência, às questões presentes no questionário, que, posteriormente será analisado com base nas mesmas.

Apresentação do cenário de teste

Cenário (Novo utilizador na aplicação)

O cenário passa pelo momento em que um novo utilizador efetua registo na aplicação e conseqüente *login*. Neste cenário será possível fazer uma gestão de atributos, exercícios e treinos, assim como visualizar o agendamento dos mesmos.

Cenário - Novo utilizador

Grupo de testes destinado aos novos utilizadores na aplicação

1. Acesso à aplicação

Aceda à aplicação clicando no seguinte endereço:
<https://smartcoachmanager.azurewebsites.net/>

Após a abertura da aplicação, efetue o registo na mesma e prossiga para o *login* com as credenciais.

2. Menu principal

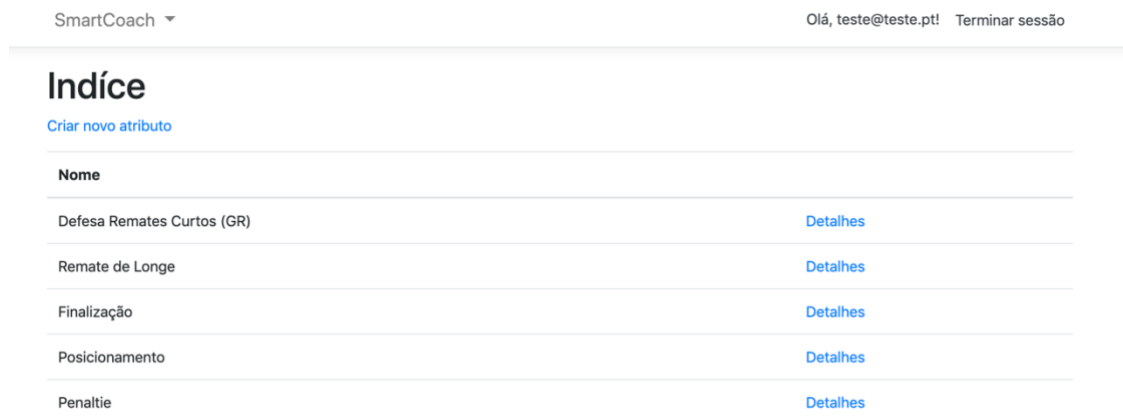
Após o login, a aplicação é redirecionada para o menu principal, onde no canto superior esquerdo irá ter o menu vertical com todas as funcionalidades que o utilizador pode escolher.



3. Gestão de Atributos

Navegue para a gestão de atributos, ao seleccionar no menu vertical a opção “**Atributos**”. Ao entrar na gestão de atributos, são listados todos os atributos já definidos no sistema.

No canto superior esquerdo, selecione a opção de criar um novo atributo.



Após o clique, irá abrir o formulário para a criação de um novo atributo. Preencha o formulário e clique no botão “Criar”.

SmartCoach ▾ Olá, teste@teste.pt! Terminar sessão

Criar

Atributo

Nome

Descrição

[Criar](#)

[Voltar](#)

4. Gestão de Exercícios

Navegue para a página de gestão de exercícios, selecionando a opção “Exercícios”.

Ao entrar na página de gestão de exercícios, são listados exercícios já definidos no sistema.

SmartCoach ▾ Olá, teste@teste.pt! Terminar sessão

Índice

[Criar novo exercício](#)

Nome	Descrição	Duração	Atributos	
2×2+1 com cruzamento e finalização	1) - Posse de bola 2×2+1 no meio com saída para os corredores com 1×1, cruzamento e finalização 2) - Após 5 passes com sucesso, jogar no colega que está no corredor 3) - Faz 1×1 dentro do quadrado e o que ganhar esse duelo faz cruzamento. Não pode ser perseguido depois de passar a linha 4) - Jogadores no meio atacam 1º poste e entrada da área. Jogador do corredor contrário ataca 2º poste 5) - Manter a rotação dos jogadores por posição	20	Finalização Cabeceamento Cruzamento Passe Curto Drible Recepção	Detalhes
2×2+1+2	1) - Posse de bola com 2×2+1, com 2 jogadores nos corredores em diagonal um em relação ao outro 2) - Nos corredores, os jogadores de fora têm 10 metros para se movimentar de uma ponta à outra e darem uma linha de passe aos jogadores de dentro 3) - 2/3 toques no máximo para os jogadores dentro 4) - 1 toque para os jogadores fora 5) - Sempre que a bola vai para o jogador exterior e volta com sucesso, ganham 1 ponto; após 20 pontos troca	15	Pressão Defensiva Recepção Passe Curto Resistência	Detalhes

Selecione a opção “Detalhes” presente em qualquer exercício listado.

Detalhes

Exercício

Nome	2x2+1 com cruzamento e finalização
Descrição	1) - Posse de bola 2x2+1 no meio com saída para os corredores com 1x1, cruzamento e finalização 2) - Após 5 passes com sucesso, jogar no colega que está no corredor 3) - Faz 1x1 dentro do quadrado e o que ganhar esse duelo faz cruzamento. Não pode ser perseguido depois de passar a linha 4) - Jogadores no meio atacam 1º poste e entrada da área. Jogador do corredor contrário ataca 2º poste 5) - Manter a rotação dos jogadores por posição
Duração (Min)	20
Imagens	



Atributos:

Nome	Classificação
Finalização	★★★★☆
Cabeceamento	★★★★☆
Cruzamento	★★★★☆
Passe Curto	★★★★☆
Drible	★★★★☆
Recepção	★★★★☆

No canto superior esquerdo da listagem, selecione a opção de criar novo exercício. Após o clique, irá abrir um formulário de criação do exercício.

Criar

Exercicio

Nome

Descrição

Duração (Min)

Vídeo (Url)

Imagens

Escolher Ficheiros Nenhum...ionado

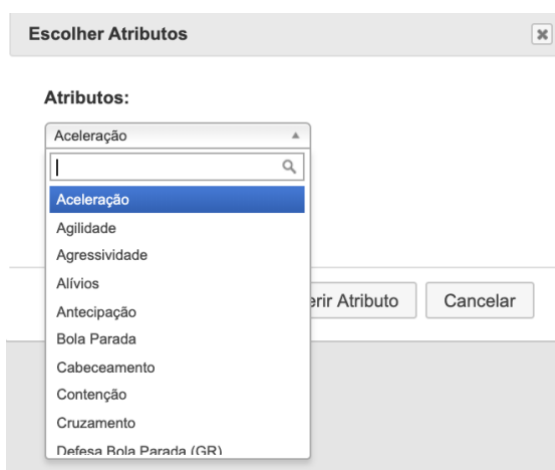
Atributos ⊞

Nome Classificação Opções

Criar

[Voltar](#)

Preencha o formulário e selecione a opção de adicionar atributos ao exercício.



Depois de adicionar um atributo, classifique-o de 0 a 5 estrelas, em que 0 o atributo selecionado não é melhorado no exercício e 5 o atributo selecionado é completamente melhorado no exercício.

Após selecionar todos os atributos desejados, selecione a opção “**Criar**”.

Criar

Exercicio

Nome

Descrição

Duração (Min)

Vídeo (Url)

Imagens

Escolher Ficheiros Nenhum...ionado

Atributos ⊞

Nome	Classificação	Opções
Aceleração	★★★★☆	Apagar

Criar

[Voltar](#)

5. Gestão de Treinos

Navegue para a página de gestão de treinos, selecionando a opção “**Treinos**”.

Ao entrar na página de gestão de treinos, não estará listado qualquer treino, visto que é um utilizador novo na aplicação e não tem nenhum treino criado.

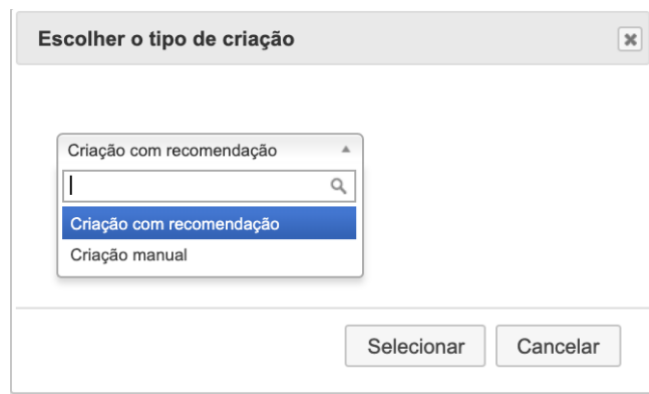
No canto superior esquerdo selecione a opção de criar novo treino.

Índice

[Criar novo treino](#)

Nome	Data	Duração (Min)	Observações	Exercícios
------	------	---------------	-------------	------------

Existem dois tipos de criação: criar treino com base numa recomendação de um sistema externo e criar um treino manual.



Selecione a opção **“Criação com recomendação”**. Após o clique, irá abrir um formulário que solicita o email do jogador que deseja obter a recomendação e o período de análise das estatísticas. Introduza os seguintes dados, visto serem dados que já existem na base de dados:

Email: maregol@fcporto.pt

Período de análise: 01/09/2019 – 30/09/2019

Após introduzir os dados, seleciona a opção **“Gerar”**.

SmartCoach ▾ Olá, teste@teste.pt! Terminar sessão

Recomendação

Informação do jogador

Email

Período de Análise

[Gerar](#)

[Voltar](#)

Após o clique, irá abrir um formulário para a criação do treino.

Criar

Treino

Nome

Data

Duração (Min)

Observações

Gerar Exercícios 

Nome	Duração (Min)	Opções
------	---------------	--------

Criar

[Voltar](#)

Preencha o formulário e selecione a opção “Gerar Exercícios”.

Criar


Treino

Nome

Data

Duração (Min)

Observações

Gerar Exercícios 

Nome	Duração (Min)	Opções
Padronizado 6x0	20	Apagar
2x2+1 com cruzamento e finalização	20	Apagar
9v9	20	Apagar

Criar

[Voltar](#)

Após gerar os exercícios para o treino, selecione a opção “**Criar**”.

De seguida, volte a repetir os passos anteriores, selecionando a opção “**Criar o treino manual**”. Após o clique, irá abrir o formulário de criação de treino manual.

SmartCoach ▾ Olá, teste@teste.pt! Terminar sessão

Criar

Treino

Nome

Data

Duração (Min)

Observações

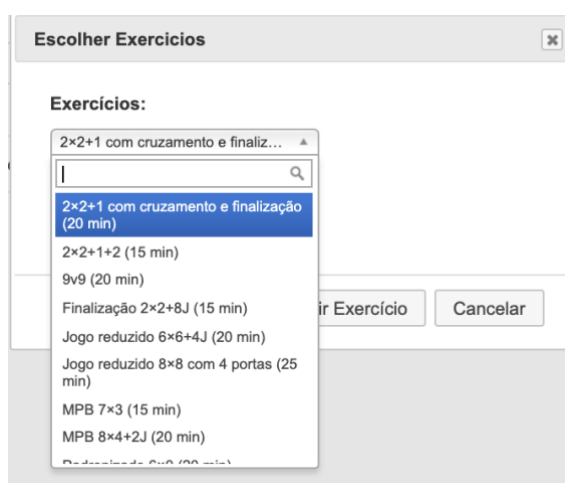
Exercícios ⊞

Nome	Duração (Min)	Opções
------	---------------	--------

[Criar](#)

[Voltar](#)

Preencha o formulário e selecione a opção de adicionar exercícios ao treino. Irá ser apresentado uma lista de todos os exercícios que existem no sistema.



Após adicionar todos os exercícios desejados, selecione a opção “**Criar**”.

Criar

Treino

Nome**Data****Duração (Min)****Observações****Exercícios** ⊞

Nome	Duração (Min)	Opções
2×2+1 com cruzamento e finalização (20 min)	20	Apagar
9v9 (20 min)	20	Apagar

[Criar](#)[Voltar](#)

6. Visualizar agendamento de treinos

Navegue para a página de visualizar o agendamento de treinos, selecionando a opção **“Planos de Treino”**.

Ao entrar na página, irá ser apresentado o calendário desportivo com todos os treinos agendados pelo utilizador.

Calendário Desportivo

< > Hoje

Outubro 2019

Mês Semana Dia Agenda

seg	ter	qua	qui	sex	sáb	dom
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

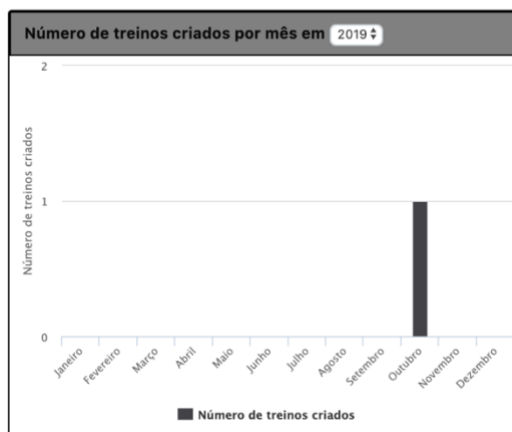
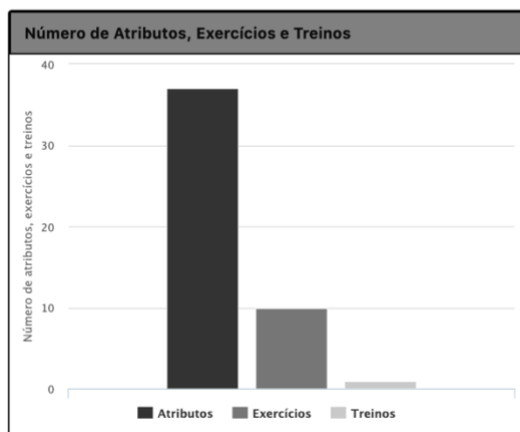
[Voltar](#)

7. Visualizar dashboard estatístico

Navegue para a página de visualizar as estatísticas da aplicação, selecionando a opção **“Dashboard”**.

Ao entrar na página, irá ser apresentado dois gráficos, o primeiro relativo ao total de atributos, exercícios e treinos que existem no sistema e o segundo relativo ao número de exercícios criados pelo utilizador.

Dashboard Estatístico



Os testes terminaram. Por favor, responda ao questionário final.

Obrigado pela sua colaboração.

Anexo 3 – Questionário

Inquérito de satisfação SmartCoach/SmartCoachManager

Este questionário é realizado de forma anónima, e tem o objetivo de recolher a opinião sobre a experiência de utilização com os sistemas SmartCoach e SmartCoachManager

*Obrigatório

1. Qual o seu cargo no clube? *

Marcar apenas uma oval.

- Treinador
- Jogador
- Membro Staff Técnico
- Outra: _____

2. A aplicação tem todas as funcionalidades desejadas? *

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

3. Considera a aplicação simples e intuitiva? *

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

4. A aplicação possui uma boa apresentação e design? *

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

5. Considera a aplicação com tempo de resposta curto? *

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

Componente SmartCoach

Plataforma associada à gestão da equipa, recolha de dados estatísticos e recomendação de treinos.

6. A gestão da equipa tem as funcionalidades pretendidas? *

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

7. Considera a configuração dos valores referência uma mais-valia? *

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

8. **Considera a recolha de estatísticas de jogo completa e intuitiva? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

9. **As recomendações de treino geradas são precisas face às necessidades do jogador? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

10. **O dashboard apresenta os gráficos/indicadores estatísticos desejados? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

Componente SmartCoachManager

Plataforma associada à gestão e classificação de treinos e exercícios de treino.

11. **A criação de exercícios de treino tem todos os parâmetros pretendidos? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

12. **Considera a classificação do exercício com base em atributos uma mais-valia? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

13. **Considera o processo de criação de um treino (manual ou por recomendação) simples e intuitivo? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

14. **Considera a calendarização dos planos de treino adequada? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

Satisfação geral

15. **A aplicação poderia ser útil na sua equipa? ***

Marcar apenas uma oval.

- Discordo completamente
- Discordo
- Sem opinião
- Concordo
- Concordo completamente

16. Como classificaria a aplicação? *

Marcar apenas uma oval.

- Muito má
- Má
- Sem opinião
- Boa
- Muito boa

17. Recomendaria a aplicação a outros utilizadores? *

Marcar apenas uma oval.

- Não
- Talvez
- Sim

18. Possíveis melhorias?
