



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Plataforma de agregação e análise de informação institucional e nas redes sociais:
uma aplicação às camadas jovens de futebol

André Santos Rouiller

Mestrado em Informática e Gestão

Orientador:

Doutor Rui Jorge Henriques Calado Lopes, Professor Auxiliar,
ISCTE-IUL

Co-Orientador:

Doutor Carlos José Corredoura Serrão, Professor Associado,
ISCTE-IUL

Outubro, 2020

Plataforma de agregação e análise de informação institucional e nas redes sociais:
uma aplicação às camadas jovens de futebol

André Santos Rouiller

Mestrado em Informática e Gestão

Orientador:

Doutor Rui Jorge Henriques Calado Lopes, Professor Auxiliar,
ISCTE-IUL

Co-Orientador:

Doutor Carlos José Corredoura Serrão, Professor Associado,
ISCTE-IUL

Outubro, 2020

Resumo

Atualmente assistimos a uma expansão do futebol português, com diversos jogadores a vingar nos melhores clubes europeus e a serem transferidos por verbas estratosféricas.

Alguma da imprensa desportiva existente, tanto em papel como em formato eletrónico, não se dedica a aprofundar o futebol nas camadas jovens por não ter uma grande visibilidade junto do público, tal como tem o futebol sénior e, hoje em dia, até mesmo o futebol feminino.

Com a identificação deste problema nas camadas jovens e a análise feita a diversos *sites*, tanto institucionais como jornalísticos e informativos, surgiu a necessidade de agregar informação dispersa sobre os jovens jogadores de futebol em Portugal.

Tendo em conta que as redes sociais estão bastante presentes no nosso quotidiano e são usadas com frequência pelos clubes, jogadores e adeptos de futebol, este tipo de informação revelou-se essencial para o desempenho dos jogadores.

Perante este contexto, foi desenvolvida uma plataforma que dará assim mais conteúdo e informação relativamente às camadas jovens de futebol, através de uma abordagem que permite a utilização da plataforma em diversos dispositivos de tamanhos diferentes.

Com os testes realizados à plataforma, concluiu-se que a agregação de informação feita é útil e aplicável às camadas jovens de futebol, garantindo em poucos segundos o acesso a dados retirados de várias fontes, o que contribui para a melhoria do futebol e dos clubes em Portugal, mais concretamente clubes amadores que não têm possibilidades financeiras para investir em tecnologia avançada de recolha de informação e análise estatística.

Palavras-chave – Futebol, Camadas jovens, Agregador de informação, Youth Football Platform

Abstract

We are currently witnessing an expansion of football in Portugal, with several key players thriving in the best European clubs and being transferred by massive compensations.

Some of the existing sports press, both in paper and electronic format, are not open to increase football knowledge among the youth players because they do not have a great visibility with the public, as does senior football and nowadays, even women's football.

With the identification of this problem among youth teams and the analysis made on several websites, both institutional and journalistic/informative, the need to aggregate dispersed information about youth football players in Portugal was detected.

Taking into account that social networks are very present in our daily lives with football clubs, players and supporters using them frequently, the information contained proved to be essential for the performance of the players.

In this context, a platform has been developed which will provide more content and information regarding youth football players, through an approach that allows the use of the platform in different devices of different sizes.

With the tests carried out on the platform, it was observed that the aggregation of the information obtained is useful and applicable to youth football players, guaranteeing almost instant access to data taken from various sources, which contributes to the improvement of football and its clubs in Portugal, more specifically, amateur clubs that do not have the financial resources to invest in advanced technology for the sourcing of information and statistical analysis.

Keywords – *Football, Youth Players, Information aggregator, Youth Football Platform*

Agradecimentos

A realização deste projeto é o resultado de muitas horas de trabalho e contou com apoios e incentivos imprescindíveis, sem os quais não teria sido possível realizá-lo e aos quais estarei eternamente grato.

Aos meus orientadores, especialmente ao Prof. Doutor Rui Jorge Lopes por todo o conhecimento transmitido, como também pela sua simpatia, compreensão, motivação, paciência, aconselhamento e disponibilidade ao longo de todo o projeto.

À minha empresa, *Axians*, que sempre demonstrou disponibilidade e abertura para que eu conseguisse conciliar o trabalho e o mestrado, podendo assim dar o melhor de mim a ambos.

À minha família e amigos pelo apoio incondicional, incentivo, amizade e paciência demonstrados ao longo desta desafiante caminhada.

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	vii
Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Acrónimos	xv
1. Introdução	1
1.1. Enquadramento.....	1
1.2. Motivação do tema	3
1.3. Objetivos	3
1.4. Metodologia e estrutura do projeto	4
1.4.1. Identificação do problema e motivação	4
1.4.2. Objetivos de uma solução	5
1.4.3. Desenho e desenvolvimento.....	5
1.4.4. Demonstração.....	5
1.4.5. Avaliação.....	5
1.4.6. Comunicação.....	6
2. Contexto: revisão da literatura e ecossistema da informação sobre futebol	7
2.1. Sites com informação desportiva	7
2.1.1. Análise das fontes de informação desportiva: relação com assistências e classificações.....	10
2.2. Serviços de anotação	13
2.2.1. Opta	13
2.2.2. InStat	15
2.2.3. Wyscout.....	18
2.3. Dispositivos comerciais de anotação e recolha de dados	19
2.3.1. STATSports.....	20
2.4. Plataformas desenvolvidas e usadas por académicos para recolha de dados	21
2.5. Impacto das redes sociais nos clubes e nos atletas.....	22
2.6. Anotação e comentário coletivo: como é que o Zerozero coleta informação?	24
2.7. Síntese da revisão da literatura.....	26

3. Desenho e desenvolvimento do sistema	27
3.1. Requisitos funcionais da plataforma	27
3.1.1. Diagrama de Casos de Uso.....	27
3.2. Arquitetura multicamada.....	30
3.3. Camadas de apresentação e lógica: Web Frameworks.....	31
3.3.1. Full-Stack Frameworks: estado da arte	32
3.3.1.1. Full-Stack Framework Django	32
3.3.1.2. Outras Full-Stack Frameworks.....	33
3.3.2. Microframeworks: estado da arte.....	33
3.3.2.1. Microframework Flask.....	34
3.3.2.2. Outras Microframeworks	34
3.3.3. Escolha de plataforma e critérios de avaliação	35
3.4. Camada de dados: tecnologias de base de dados	35
3.4.1. Bases de dados relacionais – SQL: estado da arte	36
3.4.2. Bases de dados não-relacionais – NoSQL: estado da arte	36
3.4.2.1. Base de dados MongoDB.....	39
3.4.3. Escolha da base de dados e critérios de avaliação	40
3.5. Implementação da plataforma	41
3.5.1. Desenho da base de dados.....	45
3.6. Recolha de dados.....	47
3.6.1. Recolha de dados através de Web Scraping.....	48
3.6.1.1. Procedimento de recolha de dados do site da FPF	49
3.6.1.2. Procedimento de recolha de dados do site Zerozero	50
3.6.1.3. Procedimento de recolha de dados do site O Jogo	51
3.6.2. Recolha de dados via API	51
3.6.2.1. Procedimento de recolha de dados via API do Twitter.....	52
3.6.2.2. Procedimento de recolha de dados via API do Google News.....	53
3.6.3. Recolha de dados via Selenium.....	53
3.7. Interface gráfica da plataforma	54
4. Testes e validações.....	63
4.1. Testes funcionais	63
4.2. Testes não funcionais	65
5. Conclusões e trabalho futuro	71
5.1. Conclusões	71

5.2. Trabalho futuro.....	71
Referências.....	75
Anexos	81
A. Requisitos funcionais	81
B. Vídeos dos testes realizados à plataforma	89
C. Código dos testes automatizados	92

Lista de Figuras

Figura 1. Gráfico com escala da quantidade de informação da internet por divisão	10
Figura 2. Dois anotadores da Opta a anotarem as interações dos jogadores num jogo ao vivo (fonte: Yorkshire Post).....	13
Figura 3. Jogo em direto no jornal Record, Leicester vs. Man Utd, através do "Opta match visualization"	15
Figura 4. Alex Oxlade-Chamberlain a consultar na plataforma da STATSports as métricas medidas através do APEX Pod (fonte: STATSports)	20
Figura 5. Página de colaboração do Zerozero (fonte: Zerozero).....	25
Figura 6. Página para sugerir alterações aos dados de um jogador no Zerozero (fonte: Zerozero)	25
Figura 7. Diagrama de Casos de Uso	28
Figura 8. Arquitetura de 3 camadas: apresentação, lógica e dados.....	31
Figura 9. NoSQL orientado a documento	37
Figura 10. NoSQL orientado a coluna	38
Figura 11. NoSQL orientado a grafos	38
Figura 12. NoSQL orientado a chave-valor	39
Figura 13. Diagrama de Atividades da plataforma do ponto de vista do utilizador.....	42
Figura 14. Modelo Relacional das coleções da base de dados.....	47
Figura 15. Esboço do processo de transformação da informação recolhida	48
Figura 16. Ficheiro "robots.txt" do site WhoScored.com	49
Figura 17. Parte da estrutura da plataforma que guarda os links das AF da FPF	50
Figura 18. Exemplo de utilização da API TwitterSearch (fonte: https://twittersearch.readthedocs.io/)	52
<i>Figura 19. Excerto de código da plataforma</i>	<i>53</i>
Figura 20. Exemplo de utilização do Selenium (fonte: https://www.browserstack.com/).....	54
<i>Figura 21. Página inicial da plataforma (versão móvel)</i>	<i>54</i>
Figura 22. Página inicial da plataforma (versão desktop).....	54
<i>Figura 23. Página das estatísticas</i>	<i>55</i>

<i>Figura 24. Página das estatísticas externas</i>	55
Figura 25. Página das estatísticas internas do jogador Gonçalo Batalha	56
Figura 26. Página da classificação	56
Figura 27. Página da classificação da época atual	57
Figura 28. Página da classificação do campeonato nacional de juvenis	57
Figura 29. Página da classificação das épocas anteriores	58
<i>Figura 30. Página das jornadas</i>	58
<i>Figura 31. Página da jornada 9 da 1ª divisão do campeonato nacional de júniores</i>	58
<i>Figura 32. Página inicial dos jogadores</i>	59
Figura 33. Página do jogador Gonçalo Batalha.....	59
Figura 34. Página de jogadores internos	60
<i>Figura 35. Página de autenticação para inserir um jogador na base de dados</i>	61
Figura 36. Página de registo na plataforma.....	61
<i>Figura 37. Página para inserir um jogador na base de dados da plataforma</i>	61
Figura 38. Documento JSON do jogador Gonçalo Ramos	61
Figura 39. reCAPTCHA Zerozero	64
Figura 40. Tempo médio para processar os requisitos funcionais	66
Figura 41. Interações com elementos externos da plataforma	66
Figura 42. Tempo médio para processar as estatísticas externas	68
Figura 43. Tempo médio para processar os jogadores externos	69
Figura 44. Ordem dos nomes dos jogadores procurados	92
Figura 45. Código-fonte do teste automatizado para as estatísticas externas	92
Figura 46. Código-fonte do teste automatizado para os jogadores externos.....	93
Figura 47. Código-fonte do teste automatizado para a classificação da época atual	93
Figura 48. Código-fonte do teste automatizado para a classificação de épocas passadas	94

Lista de Tabelas

Tabela 1. Sites e aplicações móveis analisados.....	8
Tabela 2. Características dos sites e aplicações móveis analisados	8
Tabela 3. Características (vídeos) dos sites e aplicações móveis analisados	9
Tabela 4. Espectadores nos estádios de clubes da 1ª Liga (fonte: LigaPortugal, 2017)	11
Tabela 5. Espectadores nos estádios de clubes da 2ª Liga (fonte: LigaPortugal, 2017)	12
Tabela 6. Características Django vs. Flask	35
Tabela 7. Lógica back-end da plataforma	44
Tabela 8. Coleção da BD – Users	45
Tabela 9. Coleção da BD – Players.....	46
Tabela 10. Requisito 1 – Obter e apresentar highlights	81
Tabela 11. Requisito 2 - Consultar estatísticas	81
Tabela 12. Requisito 3 - Consultar estatísticas externas	82
Tabela 13. Requisito 4 - Consultar estatísticas internas.....	82
Tabela 14. Requisito 5 - Consultar classificação	83
Tabela 15. Requisito 6 - Consultar classificação época atual	83
Tabela 16. Requisito 7 - Consultar classificação épocas passadas	84
Tabela 17. Requisito 8 - Consultar jornadas	84
Tabela 18. Requisito 9 - Consultar informação de jogadores	85
Tabela 19. Requisito 10 - Consultar jogadores externos.....	85
Tabela 20. Requisito 11 - Consultar informação do Twitter.....	86
Tabela 21. Requisito 12 - Consultar informação do Google News.....	86
Tabela 22. Requisito 13 - Consultar jogadores internos	87
Tabela 23. Requisito 14 - Registrar na plataforma.....	87
Tabela 24. Requisito 15 - Fazer login	88
Tabela 25. Requisito 16 - Inserir jogadores na base de dados	88

Tabela 26. Teste às funcionalidades da plataforma com base nos requisitos funcionais.....	89
Tabela 27. Teste de caixa-preta.....	89
Tabela 28. Teste automatizado às estatísticas externas.....	89
Tabela 29. Teste automatizado à classificação da época atual.....	90
Tabela 30. Teste automatizado à classificação de épocas passadas.....	90
Tabela 31. Teste automatizado aos jogadores externos.....	90
Tabela 32. Teste de caixa-branca às estatísticas externas.....	91
Tabela 33. Teste de caixa-branca à classificação da época atual.....	91
Tabela 34. Teste de caixa-branca à classificação de épocas passadas.....	91
Tabela 35. Teste de caixa-branca aos jogadores externos.....	92

Lista de Acrónimos

AF – Associação de Futebol
AJAX – Asynchronous Javascript and XML
API – Application Programming Interface
BSON – Binary JSON
DBMS – Data Base Management System
DSR – Design Science Research
DRY – Don't Repeat Yourself
FPF – Federação Portuguesa de Futebol
HTML – HyperText Markup Language
ID - Identity document
IOT – Internet Of Things
JSON – JavaScript Object Notation
ORM – Object-Relational Mapper
RDBMS – Relational Data Base Management System
RWD – Responsive Web Design
SEO – Search Engine Optimization
SQL – Structured Query Language
URL – Uniform Resource Locator
WSGI – Web Server Gateway Interface
XML – Extensible Markup Language

1. Introdução

Neste trabalho é descrito o estudo do meio envolvente, o desenho, a implementação e a avaliação de uma plataforma de agregação e análise de informação institucional e nas redes sociais sobre as camadas jovens de futebol em Portugal. É descrito todo o processo, bem como as decisões tomadas para a sua realização.

Neste primeiro capítulo é feito um enquadramento do tema abordado no trabalho com o intuito de perceber o tipo de informação que está em falta no desporto português, mais concretamente nas camadas jovens de futebol. Também é descrita a motivação que levou à realização deste trabalho, bem como os objetivos propostos e a metodologia implementada para conduzir todo o trabalho realizado.

1.1. Enquadramento

Todos sabemos que hoje em dia a informação disponibilizada na Internet é imensa e bastante mais acessível do que há 20 anos atrás, mas nem tudo é de fácil acesso, sendo que por vezes não há informação sobre o que procuramos ou há informação em excesso sem estar organizada.

Com o futebol jovem português a ser bastante falado nos últimos 6 anos, com duas conquistas nas camadas jovens, Campeonato Europeu de Futebol sub17 em 2016 e Campeonato Europeu de Futebol sub19 em 2018 (Oliveira, 2018), com quatro finais na *Youth League*¹, tendo sido uma delas ganha pelo FC Porto e as outras três perdidas pelo SL Benfica (Andrade, 2020), e com dois jogadores a serem coroados com o prémio *Golden Boy*², Renato Sanches em 2016 e João Félix em 2019 (Record, 2019), seria de esperar um maior investimento na divulgação de informação sobre as camadas jovens.

Algumas das fontes existentes não se dedicam a aprofundar o futebol nas camadas jovens por não terem uma grande repercussão junto do público, tal como tem o futebol sénior e, hoje em dia, até mesmo o futebol feminino. Com a exceção de alguns acontecimentos notáveis como os referidos anteriormente, não há o acompanhamento mais abrangente e continuado que seria expectável.

No que toca aos jornais desportivos, as notícias que mais se destacam e mais despertam interesse são as notícias relacionadas com o futebol profissional, fazendo assim com que as notícias sobre as camadas jovens não sejam a prioridade e sejam menos rentáveis para a comunicação social especializada por falta de um público alvo de grande dimensão (Araújo de

¹ https://pt.wikipedia.org/wiki/Liga_Jovem_da_UEFA

² [https://en.wikipedia.org/wiki/Golden_Boy_\(award\)](https://en.wikipedia.org/wiki/Golden_Boy_(award))

Morais, 2014). Somente quando há campeonatos europeus ou mundiais nas camadas jovens, o interesse despertado é acrescido e a visibilidade aumenta tendo uma cobertura mediática superior.

Quanto ao panorama geral dos *sites* de futebol, a informação disponibilizada sobre as camadas jovens em Portugal é bastante reduzida e por vezes nula, estando esta mesma bastante dispersa. Para fomentar esta afirmação, foi feita uma análise de vários *sites* de futebol que está inserida na secção 2.1 do capítulo “Contexto: revisão da literatura e ecossistema da informação sobre futebol”. Esta análise foi feita com base em 14 *sites* e aplicações móveis ligados ao desporto, mas com diferentes propósitos e conteúdo diverso. Inicialmente foi feita uma análise mais superficial para identificar as características de cada *site* em específico. Após identificar essas características, foram escolhidas as mais relevantes, de entre as várias características e tipos de informação sobre futebol, para fazer a análise geral e identificar as principais faltas de informação nos diversos *sites*, das quais: resultado final, relato do jogo, estatísticas do jogo, estatísticas do jogador no jogo, pontuação dos jogadores, golos, momentos, oportunidades de golo, resumos, vermelhos, amarelos e *flash interview*.

De forma especulativa e generalizada, o conteúdo nas redes sociais sobre as camadas jovens tem como público alvo principalmente os familiares dos jogadores e os próprios clubes onde estes treinam e jogam, não sendo rentável por falta de um público alvo de grande dimensão. O impacto das redes sociais nos clubes e nos atletas está descrito com maior detalhe na secção 2.5.

Com a identificação deste problema nas camadas jovens e a análise feita a diversos *sites*, tanto institucionais como jornalísticos e informativos, foi identificada a necessidade de agregar informação dispersa sobre os jovens jogadores de futebol em Portugal.

Para suprir essa necessidade, surgiu a ideia de desenvolver uma plataforma de agregação de estatísticas de equipas e jogadores nas camadas jovens de futebol em Portugal, como também informação relevante sobre os jogos, equipas e jogadores nas redes sociais, para que a informação seja mais clara, mais completa e de fácil acesso.

Um dos objetivos deste projeto é desenvolver uma plataforma com informação e estatísticas de equipas e jogadores nas camadas jovens de futebol, unindo assim dois temas bastante falados no nosso quotidiano: a tecnologia e o futebol. Isto permitirá ajudar a colmatar um *gap* de informação entre o futebol profissional e as camadas jovens que está bastante presente em Portugal. Para tornar a plataforma ainda mais informativa junta-se outro objetivo: agregar informação relevante sobre os jogadores, equipas e partidas através das redes sociais. A influência das redes sociais e o conteúdo desta mesma pode ser um fator bastante relevante no

desempenho dos jogadores e pode ter impacto direto nos resultados das equipas, sendo um tema que pode ser alvo de análise nos clubes.

Esta plataforma dará assim mais conteúdo e informação relativamente às camadas jovens de futebol, contribuindo assim para a melhoria do futebol e dos clubes em Portugal, mais concretamente clubes amadores que não têm possibilidades financeiras para investir em tecnologia avançada de análise estatística.

1.2. Motivação do tema

Para além da motivação de poder contribuir para o desenvolvimento pessoal e académico, a principal motivação é poder contribuir para o desenvolvimento do mundo da tecnologia no desporto, num sector mais específico, que é o do futebol.

Esta plataforma dá a possibilidade a diversas pessoas ligadas ao futebol de realizarem análises pormenorizadas de cada jogador, equipa, partida e também de cada torneio. Os agentes e os olheiros poderão, com a informação retirada da plataforma, realizar análises mais pormenorizadas dos jovens talentos e promissores das camadas jovens em Portugal. Os treinadores poderão melhorar o desempenho dos seus jogadores, identificando quais os pontos mais débeis e que necessitam de melhoria, assim como estar atento à interação dos jogadores nas redes sociais. Também poderão fazer uma análise dos jogos e torneios realizados, servindo assim para uma análise pré-jogo. Os árbitros poderão fazer uma análise dos jogadores mais faltosos e assim haver uma preparação antes da realização do jogo. Os jogadores poderão analisar os próprios dados estatísticos e, à semelhança do treinador, identificar os pontos que necessitam de melhoria. Os jornalistas e o público em geral também poderão retirar bastante informação da plataforma, sendo o primeiro para desenvolvimento de notícias e o segundo para conhecimento próprio.

1.3. Objetivos

Um dos objetivos deste projeto é desenvolver uma plataforma com estatísticas de equipas e jogadores nas camadas jovens de futebol. Este objetivo permite colmatar um *gap* de informação existente em Portugal entre o futebol profissional e as camadas jovens. Também irá permitir uma contribuição para o desenvolvimento dos jovens jogadores em Portugal e dos próprios clubes, maioritariamente clubes amadores sem possibilidades de investir neste ramo.

Outro dos objetivos é agregar informação relevante sobre os jogadores, equipas e partidas de futebol através das redes sociais. Este objetivo faz com que seja retirado conteúdo

dissemelhante que não se encontra nos *sites* institucionais e possibilita assim os clubes e jogadores a analisarem outro tipo de dados que, apesar de não serem adquiridos através das partidas de futebol, podem ter influência nas mesmas.

Com este projeto é pretendido que a informação recolhida e agregada de várias fontes seja fiável e certa para não gerar falsas análises, pois o impacto que pode gerar poderá ser grande, podendo ter influência diretamente no desempenho dos jogadores.

1.4. Metodologia e estrutura do projeto

A metodologia que servirá para conduzir esta investigação será a metodologia *Design Science Research* (DSR). Esta metodologia subdivide-se em seis fases (Peffer, Tuunanen, Rothenberger, & Chatterjee, 2007):

- 1) Identificação do problema e motivação
- 2) Objetivos de uma solução
- 3) Desenho e desenvolvimento
- 4) Demonstração
- 5) Avaliação
- 6) Comunicação

1.4.1. Identificação do problema e motivação

Existem algumas fontes de dados que, devido a não terem uma grande repercussão junto do público, não aprofundam o futebol nas camadas jovens como fazem com o futebol sénior e mais recentemente até o futebol feminino.

Após analisar vários *sites* informativos, institucionais e jornalísticos, e depois de identificado o problema acima mencionado nas camadas jovens, foi necessário agregar a informação dispersa sobre os jovens jogadores de futebol em Portugal.

Sendo assim, para contribuir para a melhoria do futebol e dos clubes em Portugal, mais concretamente clubes amadores que não têm possibilidades financeiras para investir em tecnologia avançada de análise estatística e recolha de informação, foi desenvolvida uma plataforma para dar mais informação relativamente às camadas jovens de futebol.

Esta fase da identificação do problema e motivação é abordada com maior detalhe no Capítulo 2 - Contexto: revisão da literatura e ecossistema da informação sobre futebol.

1.4.2. Objetivos de uma solução

Esta plataforma irá permitir obter mais informação sobre as equipas e jogadores das camadas jovens de futebol em Portugal, disponível para ser acedida por qualquer pessoa em qualquer lugar e com a garantia de que é de fácil compreensão e utilização.

Também será bastante importante para diversas pessoas, como é o caso dos jogadores, que irão ganhar outra visibilidade consoante a sua produção em campo, dos olheiros, que com base nas estatísticas conseguem ver, por exemplo, o melhor marcador de uma certa faixa etária, dos treinadores e dos clubes de formação pois elevam o estatuto dentro dos clubes formadores em Portugal. A utilização de estatísticas de futebol tem contribuído para o melhoramento dos aspetos técnicos do jogo, tanto a nível individual como coletivo (Vendite, Moraes, & Vendite, 2015). Esta fase está descrita com maior detalhe no início do Capítulo 3 - Desenho e desenvolvimento do sistema.

1.4.3. Desenho e desenvolvimento

Segundo March e Smith (1995), uma instanciação é a concretização de um artefacto no seu ambiente, e, sendo neste caso o artefacto uma plataforma, o ambiente utilizado para o seu desenvolvimento será o ambiente *web*. A abordagem seguida é a abordagem *RWD (Responsive Web Design)*, que faz com que as páginas *web* sejam adaptadas a qualquer dispositivo existente.

Nesta fase será feito o desenvolvimento da plataforma de agregação e análise de informação institucional e nas redes sociais, sendo demonstrada e descrita no Capítulo 3 - Desenho e desenvolvimento do sistema.

1.4.4. Demonstração

O ponto 4 da metodologia servirá para simular a utilização do artefacto através de testes realizados à plataforma desenvolvida, procurando assim resolver os problemas identificados nesta investigação. Esta fase é descrita com maior detalhe no Capítulo 4 - Testes e validações.

1.4.5. Avaliação

A fase de avaliação “é definida como o processo rigoroso de verificação do comportamento do artefacto no ambiente para o qual foi projetado” (Lacerda, Dresch, Proença, & Antunes Júnior, 2013). De forma a validar o artefacto e o desempenho deste mesmo, serão feitos dois tipos de testes estruturais e funcionais: automatizados e manuais.

Este tipo de avaliação será um processo para descobrir erros de requisitos, design e codificação do programa, e portanto será utilizada a técnica *white box* (caixa branca) e *black box* (caixa preta) (Ehmer & Khan, 2012).

A vantagem de ter estes dois tipos de testes (automatizados e manuais) é que os testes automatizados podem ser feitos com maior frequência e mais rapidamente sempre que for necessário (Correia & da Silva, 2004) através de uma máquina, enquanto que os testes manuais são feitos por seres humanos passo-a-passo para obter o resultado esperado, podendo ser alterados a meio do processo de teste. Esta fase é descrita com maior detalhe no Capítulo 4 - Testes e validações.

1.4.6. Comunicação

A fase de comunicação serve para comunicar o problema e a sua importância, a utilidade e a novidade do artefacto, o rigor e o seu design e a sua efetividade para investigadores e outras audiências relevantes (Peppers et al., 2007).

Fala-nos também das estratégias para disseminar o trabalho realizado: a escrita do projeto e a escrita de artigos científicos para descrever o trabalho realizado.

Com a identificação do problema de falta de informação nas camadas jovens e a análise feita a diversos *sites*, tanto institucionais como jornalísticos e informativos, foi detetada a necessidade de agregar informação dispersa sobre os jovens jogadores de futebol em Portugal. Para solucionar este problema, foi desenvolvida uma plataforma intuitiva que dará assim mais conteúdo e informação relativamente às camadas jovens de futebol, contribuindo para a melhoria do futebol e dos clubes em Portugal, mais concretamente clubes amadores que não têm possibilidades financeiras para investir em tecnologia avançada de análise estatística. A única documentação do trabalho realizado foi este trabalho de projeto de mestrado.

2. Contexto: revisão da literatura e ecossistema da informação sobre futebol

Só nos últimos anos é que os dados estatísticos de futebol se desenvolveram, graças às tecnologias de detecção que fornecem fluxos de dados de alta fidelidade extraídos de cada partida de futebol (Pappalardo et al., 2019).

Dentro destas tecnologias é possível identificar dois tipos: os serviços de anotação e os dispositivos comerciais de anotação e recolha de dados. Ambos os tipos têm uma base de dados com informação relacionada com futebol, mas a principal diferença é que os dispositivos de anotação, para além de terem um *software* incorporado como os serviços de anotação, também têm produtos físicos associados. Enquanto que os serviços de anotação estão focados em retirar dados estatísticos dos jogos através de vídeo (Figura 2), os dispositivos de anotação preocupam-se mais com o desempenho e índices físicos dos jogadores antes (nos treinos) e durante o jogo através de produtos físicos (Figura 4) que geram dados para serem vistos através de um *software*.

Neste capítulo, para além do que foi referido anteriormente, também é feita uma análise de *sites* com informação desportiva, é abordado o que falta neste ecossistema da informação sobre futebol e onde há a possibilidade de ter impacto, são dados exemplos de plataformas criadas e usadas por académicos para recolha de dados, é abordado o impacto das redes sociais nos clubes e nos atletas e, por último, será abordada a forma como o *Zerozero* coleta informação para a sua base de dados.

2.1. Sites com informação desportiva

A escolha dos *sites* para fazer uma análise relativamente à informação sobre as camadas jovens de futebol foi feita com base em diferentes tipos de fonte e diversos *rankings*. Dos *sites* de informação institucional fazem parte diversas fontes oficiais de desporto em Portugal: os três principais jornais desportivos em Portugal (*A Bola*, *Record* e *O Jogo*), o *site* da FPF e o *site* da *Liga Portugal*. Quanto à informação não institucional, foram escolhidos diversos *sites* de informação desportiva presentes em diversos *rankings*. Segundo o *ranking* dos *sites* mais visitados em Portugal em Novembro de 2020 (SimilarWeb, 2020a), o *Zerozero* consta no topo da lista. O *ranking* dos 50 *sites* mais visitados sobre futebol a nível mundial (SimilarWeb, 2020b) apresenta na sua lista o *LiveScore*, *FlashScore*, *Transfermarkt* e o *WhoScored*. Para além de aparecer nos *rankings* da *SimilarWeb*, o *Transfermarkt*, o *FlashScore* e o *WhoScored*

também aparecem em diversos *rankings* dos melhores *sites* de análise estatística de futebol, como é o caso *Football-Index* (2020). do *The Punters Page* (2020) e do *Aelieve* (2019).

	Link	Tipo	Casa de apostas associada	Entrevistas
A Bola	www.abola.pt	Site	Sim	Sim
FlashScore	www.flashscore.com	Site / APP	Não	Não
FootLive	www.footlive.com	Site	Não	Não
Forza Football	www.forzafootball.com	APP	Sim	Não
FPF	www.fpf.pt	Site	Sim	Sim
Futebol365	www.futebol365.pt	Site	Sim	Não
Liga Portugal	www.ligaportugal.pt	Site / APP	Sim	Sim
LiveScore	www.livescore.com	Site / APP	Sim	Não
O Jogo	www.ojogo.pt	Site	Sim	Sim
Record	www.record.pt	Site / APP	Sim	Sim
Transfermarkt	www.transfermarkt.pt	Site / APP	Não	Não
WhoScored	www.whoscored.com	Site / APP	Sim	Sim
WyScout	www.wyscout.com	Site / APP	Não	Não
Zerozero	www.zerozero.pt	Site	Sim	Sim

Tabela 1. Sites e aplicações móveis analisados

	Características				
	Resultado final	Relato do jogo	Estatísticas do jogo	Estatísticas do jogador no jogo	Pontuação dos jogadores
A Bola	I(P) - 1D - 2D	I(P) - 1D	X	X	X
FlashScore	I - 1D - 2D	I - 1D(P)	I - 1D - 2D	X	X
FootLive	I - 1D - 2D	X	X	X	X
Forza Football	I - 1D - 2D - CJ(P)	I - 1D - 2D	I - 1D - 2D	X	X
FPF	1D - 2D - CJ	X	X	X	X
Futebol365	I - 1D - 2D - CJ	X	X	X	X
Liga Portugal	1D - 2D	1D - 2D	1D - 2D	X	X
LiveScore	I - 1D - 2D	I - 1D - 2D	I - 1D - 2D	X	X
O Jogo	I(P) - 1D	I(P) - 1D	X	X	X
Record	I - 1D - 2D - CJ	I - 1D	I - 1D	X	X
Transfermarkt	I - 1D - 2D	I - 1D - 2D	I	X	X
WhoScored	I - 1D - 2D	I - 1D	I - 1D	I - 1D	I - 1D
WyScout	I - 1D	X	I - 1D	I - 1D	X
Zerozero	I - 1D - 2D - CJ	I - 1D - 2D - CJ(P)	I - 1D(P)	I	X
Legenda:	X - Não tem	I - Tem (Internacional)	1D - Tem (1ª Divisão PT)	2D - Tem (2ª Divisão PT)	
	CJ - Tem (Camadas Jovens)	(P) - Pouca informação	(R) - É preciso registar	* - Desaparece após 2 dias	

Tabela 2. Características dos sites e aplicações móveis analisados

Na Tabela 1 estão os *sites* e aplicações móveis que foram analisados, bem como se estes mesmos têm casa de apostas associada e se têm entrevistas exclusivas.

Na Tabela 2 estão várias características escolhidas para avaliar os *sites/aplicações* móveis. A coluna “Resultado final” representa se o *site* exibe o resultado do jogo ocorrido, a coluna “Relato do jogo” representa se o *site* exibe um relato escrito do jogo, a coluna “Estatísticas do jogo” representa se o *site* exibe estatísticas (tais como posse de bola e remates) do jogo ocorrido, a coluna “Estatísticas do jogador no jogo” representa se o *site* exibe estatísticas (tais como distância percorrida e passes realizados) dos jogadores que participaram no jogo, e por último a coluna “Pontuação dos jogadores” representa se o *site* exibe uma pontuação final ao desempenho do jogador na partida.

	Vídeos						
	Golos	Momentos	Oportunidades de golo	Resumos	Vermelhos	Amarelos	Flash Interview
A Bola	I(P) - 1D	I(P) - 1D	X	1D	X	X	X
FlashScore	I - 1D *	X	X	X	I - 1D *	X	X
FootLive	X	X	X	X	X	X	X
Forza Football	I - 1D	X	X	I - 1D	X	X	X
FPF	X	X	X	X	X	X	X
Futebol365	X	X	X	I - 1D - 2D (P) - CJ(P)	X	X	X
Liga Portugal	1D	1D	1D	1D	1D	X	1D
LiveScore	X	X	X	X	X	X	X
O Jogo	I(P) - 1D(P)	I(P) - 1D(P)	X	X	X	X	X
Record	I - 1D	1D	1D	I - 1D	1D	X	1D
Transfermarkt	X	X	X	X	X	X	X
WhoScored	X	X	X	X	X	X	X
WyScout	I - 1D	I - 1D	I - 1D	I - 1D	I - 1D	I - 1D	X
Zerozero	I - 1D	I - 1D	I - 1D	I - 1D - CJ(P)	I - 1D	X	I - 1D
Legenda:	X - Não tem		I - Tem (Internacional)	1D - Tem (1ª Divisão PT)		2D - Tem (2ª Divisão PT)	
	CJ - Tem (Camadas Jovens)		(P) - Pouca informação	(R) - É preciso registar		* - Desaparece após 2 dias	

Tabela 3. Características (vídeos) dos sites e aplicações móveis analisados

Como é possível verificar na Tabela 2 e na Tabela 3, há poucas plataformas/*sites* com informação sobre as camadas jovens, sendo que em alguns *sites* esta mesma é nula. Muitos destes *sites* que contêm informação sobre camadas jovens têm informação desatualizada ou bastante incompleta, o que gera um problema para quem procura este tipo de informação.

Nenhum dos *sites*/aplicações móveis analisados é totalmente vocacionado para as camadas jovens e embora tenha sido pesquisado, não foi encontrado nenhum *site*/aplicação móvel deste género em Portugal.

A motivação desta plataforma é poder fazer algo nas áreas do futebol e da informática, resolvendo um problema existente devido a um *gap* de informação no futebol que pode ser colmatado com a ajuda desta mesma.

2.1.1. Análise das fontes de informação desportiva: relação com assistências e classificações

Quando procuramos informações sobre futebol profissional sénior e camadas jovens, podemos rapidamente perceber que há uma disparidade muito grande em termos de informação.

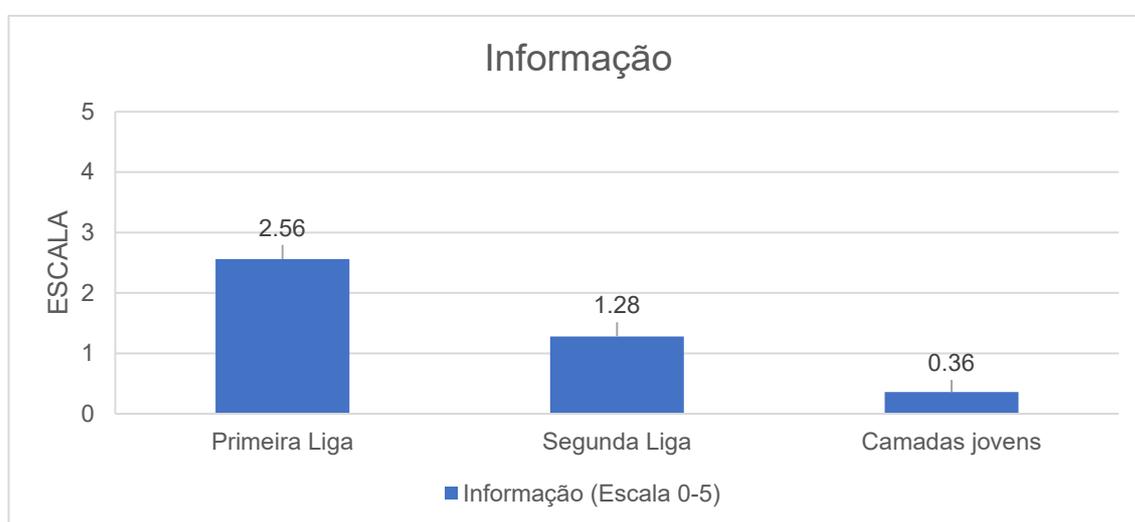


Figura 1. Gráfico com escala da quantidade de informação da internet por divisão

O gráfico acima demonstra a diferença de informação existente na *internet* relativamente à primeira liga, segunda liga e camadas jovens em território português. Ou seja, numa breve análise conseguimos observar naturalmente que, consoante a divisão é mais alta, maior interesse desperta diante do público.

Esta escala foi feita com base na análise de características (resultado final, relato do jogo, estatísticas do jogo, estatísticas do jogador no jogo, pontuação dos jogadores e vídeos) de 14 *sites* diferentes feita na secção 2.1, sendo 0 (Nenhuma informação) até 5 (Muita informação).

O melhor *site* para informação sobre a Primeira Liga é o *WhoScored*, pois apesar de não conter vídeos dos jogos contém bastante informação estatística. Já na Segunda Liga, existem três *sites* que se destacam: *Forza Football*, *Liga Portugal* e *LiveScore*, com a particularidade de nenhum dos três conter vídeos dos jogos. Por último, nas camadas jovens, o *site* com mais

informação é o *Zerozero* sendo que, apesar de ser o *site* com mais informação, numa escala de 0-5 apenas obtém 1,67.

Esta falta de informação pode ter inúmeros motivos, de entre os quais a falta de interesse por parte do público e consequentemente por parte dos *sites* e a falta de recursos/meios.

Relativamente à perda de interesse do público à medida que a divisão é menor, foi feito um estudo relativamente aos espectadores da temporada 2018/2019 da Primeira Liga e Segunda Liga que assistiram aos jogos no estádio ao longo da época.

	Classificação final	Espectadores 1ª liga - Época 18/19 *	
		Média	Total
SL Benfica	1º	53 824	915 003
FC Porto	2º	41 626	707 650
Sporting CP	3º	33 691	572 743
Vitória SC	5º	18 249	310 240
SC Braga	4º	12 035	204 598
Boavista FC	8º	8 155	138 639
Marítimo	12º	6 622	112 576
Vitória FC	13º	4 784	81 336
GD Chaves	16º	4 550	77 349
Santa Clara	10º	4 010	68 162
Rio Ave FC	7º	3 630	61 712
Portimonense	11º	3 313	56 320
CD Feirense	18º	3 049	51 834
Belenenses	9º	2 889	49 114
CD Tondela	15º	2 702	45 930
CD Nacional	17º	2 595	44 118
CD Aves	14º	2 454	41 726
Moreirense	6º	2 275	38 670

Tabela 4. Espectadores nos estádios de clubes da 1ª Liga (fonte: LigaPortugal, 2017)

Como podemos ver na Tabela 4, relativamente à média de espectadores por jogo, conseguimos repartir os 18 clubes por 3 níveis: o 1º nível composto por apenas três clubes que conseguem ter uma média superior a 30 mil espectadores por jogo (SL Benfica, FC Porto e Sporting CP); o 2º nível composto por dois clubes que conseguem entre 10 e 20 mil espectadores (Vitória SC e SC Braga); e o 3º nível composto pelos restantes 13 clubes que colocam menos de 10 mil espectadores por jogo (Boavista FC, Marítimo, Vitória FC, GD Chaves, Santa Clara, Rio Ave FC, Portimonense, CD Feirense, Belenenses, CD Tondela, CD Nacional, CD Aves, Moreirense).

	Classificação final	Espetadores 2ª liga - Época 18/19 *	
		Média	Total
FC Famalicão	2º	3 478	59 133
FC P. Ferreira	1º	2 772	47 129
A. Académica	5º	2 631	44 722
SC Farense	10º	2 437	41 423
Varzim SC	15º	1 606	27 305
Vitória SC B	18º	1 035	17 597
FC Arouca	16º	976	16 586
CD C. Piedade	13º	965	16 406
Estoril Praia	3º	895	15 212
Leixões SC	7º	895	15 207
Ac. Viseu	11º	724	12 310
SL Benfica B	4º	657	11 167
CD Mafra	14º	617	10 482
FC Penafiel	8º	594	10 097
FC Porto B	9º	488	8 302
UD Oliveirense	12º	486	8 262
SC Covilhã	6º	461	7 830
SC Braga B	17º	337	5 724

Tabela 5. Espectadores nos estádios de clubes da 2ª Liga (fonte: LigaPortugal, 2017)

Na Tabela 5, relativamente ao total de espectadores da Segunda Liga, podemos repartir os 18 clubes por 2 níveis: o 1º nível composto por seis clubes que conseguem ter mais de mil espectadores por jogo (FC Famalicão, FC P. Ferreira, A. Académica, SC Farense, Varzim SC e Vitória SC B); e o 2º nível composto por doze clubes que colocam menos de mil espectadores por jogo (FC Arouca, CD C. Piedade, Estoril Praia, Leixões SC, Ac. Viseu, SL Benfica B, CD Mafra, FC Penafiel, FC Porto B, UD Oliveirense, SC Covilhã e SC Braga B).

Também é observável que a classificação final do clube não está ligada com a quantidade de espectadores no estádio, havendo dois casos mais peculiares tanto na Primeira Liga como na Segunda Liga:

- Na Primeira Liga (Tabela 4), o GD Chaves desceu de divisão, no entanto foi o 9º clube com maior assistência da Primeira Liga; e o Moreirense, que ficou no 6º lugar na classificação final, teve a pior assistência da Primeira Liga;
- Na Segunda Liga (Tabela 5), o SC Covilhã acabou em 6º lugar na classificação final, no entanto é o 2º pior clube em termos de assistência; o Vitória SC B desceu de divisão mas conseguiu a 6ª melhor assistência da Segunda Liga.

Com estas duas tabelas (Tabela 4 e Tabela 5), verifica-se uma descida de interesse do público consoante a divisão. O volume de espectadores nos estádios da Primeira Liga é bastante superior ao da Segunda Liga, e com isso surge uma justificação para uma menor quantidade de notícias e estatísticas relativamente à Segunda Liga.

2.2. Serviços de anotação

Annotation sources, ou serviços de anotação, são serviços que se dedicam a analisar e gerar dados estatísticos desportivos. Estes dados não são de livre acesso (Pappalardo et al., 2019), sendo necessário pagar para obtê-los, dependendo este valor da magnitude de informação pretendida por quem quer obter estes dados.

Como exemplos de serviços de anotação existem, por exemplo, a *Opta*³, a *InStat*⁴ e a *WyScout*⁵, três empresas de dados desportivos.

2.2.1. Opta

A *Opta* é uma empresa de dados desportivos que analisa, armazena e distribui esses mesmos dados para milhares de dispositivos em todo o mundo. Estes dados são utilizados em várias indústrias, tais como: apostas, imprensa, patrocínio, ativação da marca, televisão e análise de desempenho profissional (Opta, 1996).



Figura 2. Dois anotadores da Opta a anotarem as interações dos jogadores num jogo ao vivo (fonte: Yorkshire Post).

³ <https://www.optasports.com>

⁴ <https://instatsport.com>

⁵ <https://wyscout.com>

Os dados da *Opta* são bastante detalhados e tanto são ao vivo (dados atualizados a cada segundo de jogos que estão a decorrer) (Figura 2), como histórico (dados de jogos finalizados). Esta empresa trabalha em colaboração com vários clientes para fornecer um serviço cada vez melhor, tanto para o cliente como para a *Opta* (Opta, 1996).

Relativamente ao estudo feito a vários *sites* (Tabela 1) (Tabela 2) (Tabela 3), o *WhoScored*⁶ e o jornal *Record*⁷ são um exemplo de utilizador desta fonte de dados. Utilizam nos jogos em direto, mais concretamente nos indicadores do desempenho do jogador, relato do jogo e estatísticas do jogo. Para além desta utilização dos dados da *Opta*, o *WhoScored* também utiliza esta fonte para estatísticas detalhadas dos artigos de especialistas.

Relativamente aos serviços prestados, existem cinco opções: *data feeds*, *widgets*, *editorial*, *analytics* e *opta match visualisation* (Opta, 1996).

A. *Data feeds*

De entre as opções, esta é a opção mais flexível, mas também a mais complexa. Estes dados são capturados em tempo real antes de serem entregues aos clientes em forma de código (*XML*). Esta opção está dividida em 4 níveis: *core*, *classic*, *performance* e *advanced*. O nível um (*core*) é de resultados ao vivo, atualizações e conteúdo pós-jogo. O nível dois (*classic*) engloba dados e estatísticas mais detalhadas sobre jogadores e equipas. O nível três (*performance*) é o nível mais detalhado e preciso relativamente ao desempenho global e individual dos jogadores/equipa. Por último, o nível quatro (*advanced*) refere-se a dados derivados de modelagem e avaliação analítica.

B. *Widgets*

Serve para apresentar dados estatísticos de jogos já realizados e é uma forma bastante menos complexa do que *data feeds*, podendo ser customizado à medida do cliente. Pontuações do desempenho de jogadores nas partidas, hipóteses de golo da época e zonas defensivas são exemplos de *widgets* possíveis desenvolvidos pela *Opta*.

C. *Editorial*

Para os jornais de notícias, esta é a opção mais acertada em termos de histórias e conteúdo exclusivo. Sabendo do que é que o público mais gosta de ler, fazem diversas análises específicas, dando ao cliente conteúdo que pode ser publicado.

⁶ <https://www.whoscored.com>

⁷ <https://www.record.pt/>

D. Analytics

Esta opção permite obter dados para análise do cliente, e um exemplo deste tipo de dados é um gráfico de barras “número de golos marcados vs número de golos expectáveis”. O desenvolvimento de modelos preditivos (feitos com dados históricos detalhados) e métricas analíticas (para estudar melhor o desempenho dos jogadores e da equipa) são outros exemplos desta opção de dados.

E. Opta match visualisation

Esta é a opção mais recente, mais cativante e com maior margem de crescimento. Como o nome indica, serve para acompanhar e visualizar dados das partidas de futebol ao vivo, conforme o desenvolvimento das ações dentro desta mesma. Mostra as localizações dentro do campo de todas as ações dos jogadores em tempo real, incluindo passes, remates, livres, faltas, golos e ataques. Um exemplo de cliente que utiliza este tipo de serviço, para o campeonato português, é o jornal *Record*.

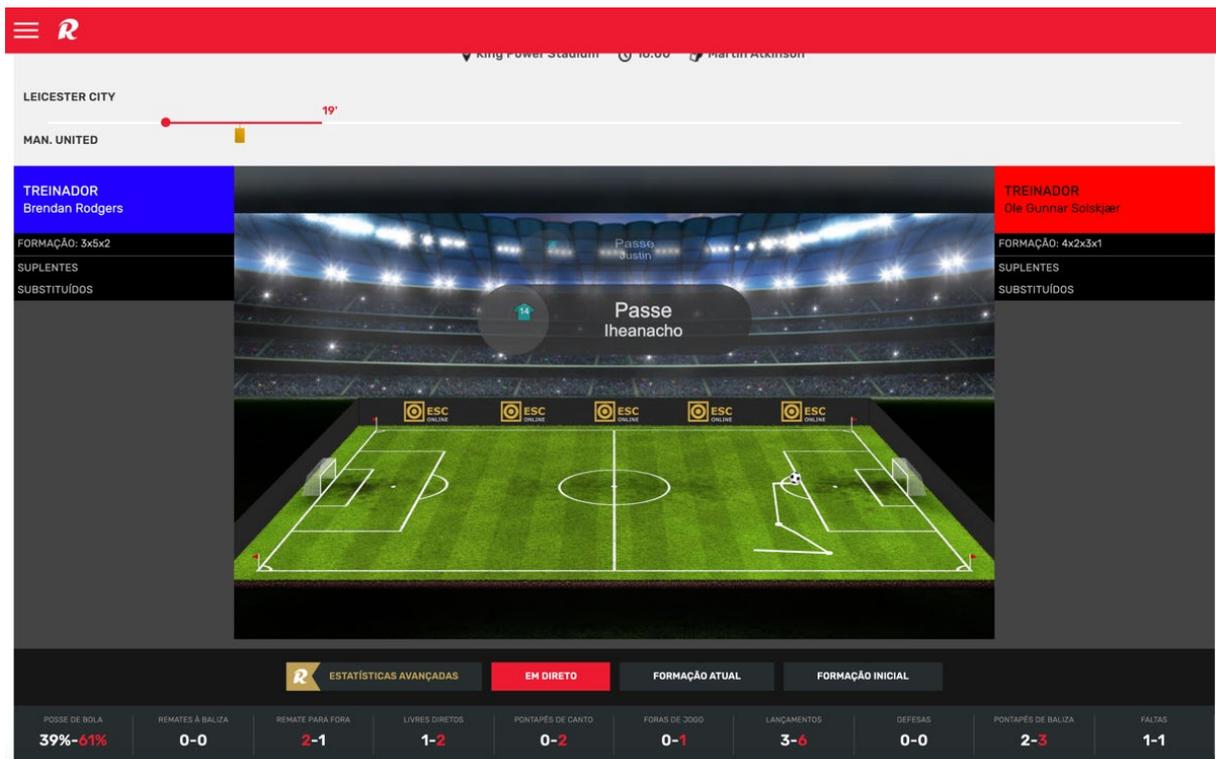


Figura 3. Jogo em direto no jornal Record, Leicester vs. Man Utd, através do "Opta match visualization"

2.2.2. InStat

A *InStat* é uma empresa de análise de desempenho desportivo, que fornece ferramentas profissionais para avaliação de desempenho individual e de equipa, *scouting*, análise de condicionamento físico e uma tecnologia de filmagem panorâmica (InStat, 2007).

A *InStat* é diferente da *Opta* pois o principal objetivo é dar apoio aos profissionais de futebol com dados estatísticos.

Quanto aos serviços prestados pela *InStat*, existem 10 opções: *InStat Scout*, *Statistical reports*, *Analytical reports*, *inStat Autocrop*, *inStat Index*, *For Academies*, *For Players*, *Scouting reports*, *inStat Fitness* e *For Referees* (InStat, 2007).

A. *InStat Scout*

Este serviço possibilita treinadores, jogadores, olheiros, agentes, jornalistas e outros profissionais do futebol a visualização de vídeos, estatísticas e gráficos interativos sobre mais de 960 000 jogadores. Um dos clientes mais ativos desde 2016 deste tipo de serviço é a Federação Francesa de Futebol, que em 2016 foi vice-campeã europeia e em 2018 sagrou-se campeã do mundo.

B. *Statistical reports*

Após o apito final de cada jogo, o relatório da equipa, o relatório do jogador, o relatório resumido do jogador e o relatório do torneio chegam ao utilizador. Este é o principal produto da *InStat*, com clientes de grande nome a terem este serviço, como é o caso do Manchester United, Liverpool, Barcelona, PSG, Federação Francesa de Futebol e a Federação Russa de Futebol.

C. *Analytical reports*

Este serviço fornece uma análise tática da partida, determina padrões num relatório estatístico e encontra tendências. Este tipo de relatório contém informações estatísticas e podemos analisar as partidas anteriores e os próximos adversários. Durante o Euro 2016, foi pedido à *InStat* para analisar os adversários da Islândia e no Mundial de 2018 analisou o estilo de jogo da Espanha e os batedores de penaltis da Croácia a pedido da Rússia.

D. *InStat Autocrop*

Permite ver todos os jogadores e analisar o desempenho da equipa. Este serviço permite filmar partidas de futebol sem a necessidade de um operador de câmara, somente será preciso que alguém configure uma câmara panorâmica antes do início de cada partida.

E. InStat Index

É desenvolvida através de um algoritmo que mede a contribuição do jogador para o sucesso do clube, o significado das suas ações, o nível do adversário e o nível do campeonato onde joga. Esta opção ajuda a determinar a classe do jogador e com isso o utilizador será capaz de perceber se um jogador suplente está pronto para jogar como titular, quão difícil será transferi-lo para uma liga superior e o desempenho do jogador em partidas de grau de dificuldade diferente. Treinadores como Carlo Ancelotti e Rafael Benítez utilizam esta ferramenta, assim como jogadores, dando o exemplo de Antoine Griezmann, Artem Dzyuba, Petr Cech e Karim Benzema.

F. For Academies

Esta ferramenta é para analisar uma faixa etária restrita entre os 13 e os 18 anos de idade. São gerados relatórios estatísticos, resumos em vídeo, análise de pontos fortes e inclusão dos jogadores no banco de dados *InStat*. Estes dados são gerados com base em torneios juvenis e juniores, contando hoje em dia com informações sobre mais de 45 000 jogadores com idades entre os 13 e os 18 anos. O uso destes dados permite que os treinadores das academias vejam os pontos fortes dos jogadores e direcionem mais acertadamente o seu desenvolvimento. Este tipo de serviço é utilizado por grandes clubes europeus, como é o caso do Real Madrid, Atlético de Madrid e Manchester City.

G. For Players

Como o nome indica, esta opção é dedicada aos jogadores. Relatórios individuais após cada partida realizada, vídeos de partidas completas e ações de jogadores específicos. Esta ferramenta é bastante utilizada por profissionais para melhorar o seu rendimento e estilo de jogo, corrigindo falhas que podem ser visualizadas por esta mesma. Também pode servir para um jogador analisar os oponentes do próximo jogo, fazendo assim com que haja uma preparação pré-jogo bastante completa. Um exemplo de utilizador desta ferramenta é Antoine Griezmann, jogador do Barcelona e campeão do Mundo pela França.

H. Scouting reports

Este serviço é para olheiros e todos os envolvidos na transferência de jogadores. Este tipo de relatório pode avaliar um jogador específico, selecionar uma quantidade de jogadores para uma posição específica, entre outros.

I. InStat Fitness

Esta é uma ferramenta bastante útil para os treinadores e jogadores, uma vez que a condição física é um fator que influencia bastante o desempenho individual/coletiva. Este serviço é entregue em forma de relatório com a atividade física da equipa e dos jogadores em cada partida, como a distância percorrida, velocidade atingida, entre outros. Esta opção é disponibilizada ao cliente horas depois do fim da partida e um dos principais clientes deste serviço é, a nível nacional, o FC Porto;

J. For Referees

Como o nome indica, esta opção é dedicada aos árbitros. É o estudo do desempenho dos árbitros com mais detalhe, via relatório, depois de cada jogo arbitrado. Estas estatísticas ajudam a perceber melhor a importância dos árbitros no desenrolar do jogo e quão bem acompanharam a partida. Quanto à veracidade da informação demonstrada no relatório, pode ser confirmada via vídeo, também disponibilizado pela *InStat*.

2.2.3. Wyscout

A *Wyscout* é a empresa líder mundial em informação para *scouting* de jogadores e análise de partidas de futebol, tendo na sua base de dados milhares de perfis de jogadores, vídeos, estatísticas e relatórios. Fornece um conjunto de ferramentas criadas especificamente para profissionais do futebol, tais como agentes, olheiros, treinadores, árbitros, jogadores de futebol e jornalistas (Wyscout, 2004).

Esta plataforma foi concebida para quem tem a necessidade de ter uma melhor visão do que aconteceu num jogo específico e também para quem quer analisar jogadores de futebol ao pormenor com diversas estatísticas personalizadas.

Os serviços disponíveis na plataforma da *Wyscout* são os seguintes: vídeos, dados e estatísticas, produtos especiais e produtos extra (Wyscout, 2004).

A. Vídeos

Esta funcionalidade da plataforma permite aos utilizadores descarregar vídeos e fazer uma análise de vídeo de partidas de futebol. São carregados semanalmente mais de 2000 novos jogos de futebol na plataforma, desde as melhores ligas europeias a torneios juvenis mais importantes do mundo, contabilizando mais de 250 competições de futebol.

B. Dados e estatísticas

Este serviço permite obter informações atualizadas sobre jogos de futebol, jogadores, equipas e competições de futebol, sendo adequado para quem pretende ter uma visão mais aprofundada sobre o que procura. São disponibilizadas estatísticas bastante detalhadas, tendo a possibilidade de obter um relatório profissional.

C. Produtos especiais

No que toca aos produtos especiais, estes são compostos por ferramentas que possibilitam descobrir jogadores com capacidades específicas, personalizar vídeos com ferramentas de edição especiais, criar classificações personalizadas de jogadores, planear a construção de um plantel, promover jogadores durante o mercado de transferências e criar listas personalizadas de jogadores.

D. Produtos extra

Para além destes serviços, a *Wyscout* também tem produtos extra para aceder a listas dos melhores talentos no futebol com menos de 21 anos de idade, analisar jogadas ensaiadas, fazer relatórios personalizados de jogadores e visualizar vídeos e estatísticas sobre árbitros.

Quanto à falta de recursos/meios, muitos clubes não têm a possibilidade de ter acesso a serviços de anotação porque, apesar de haver inúmeras soluções, todas elas têm como target os clubes profissionais de futebol devido ao elevado custo destas mesmas (Barros, Serrão, & Lopes, 2018), e com isso recorrem a outras alternativas, que apesar de não serem tão completas poderão ajudar a atingir os objetivos do clube (MaisFutebol, 2016)(Bancada, 2019). Uma dessas alternativas é o *Football Manager*, série de jogos eletrónicos em que o objetivo principal é treinar uma equipa de futebol. Neste jogo conseguimos informação sobre vários jogadores e o potencial que poderão atingir no futebol (Football Manager, 2005).

2.3. Dispositivos comerciais de anotação e recolha de dados

Annotation Tech Producers, ou produtores de tecnologia de anotação, são os fornecedores de equipamentos de modo a obter dados estatísticos de jogadores.

Como exemplos de *annotation tech producers* existe, por exemplo, a *STATSports*⁸:

⁸ <https://statsports.com>

2.3.1. STATSports

A *STATSports* é o principal fornecedor de equipamentos de rastreamento e análise de jogadores em desportos de elite, como é o caso do futebol, basquetebol, futebol americano, rugby e atletismo (STATSports, 1981).

O sistema *STATSports APEX/Viper* é uma plataforma orientada ao cliente. A interface deste *software* foi projetada e modelada juntamente com alguns dos melhores treinadores do mundo, departamento médico dos clubes e treinadores de desporto (STATSports, 1981).

Este sistema é suportado por um dispositivo, de seu nome *APEX Pod (APEX Athlete Series* ou *APEX Pro Series)*, que é colocado num colete (*Vest*) ou numa camisola *long sleeve (Base Layers)*, ilustrado na Figura 4.



Figura 4. Alex Oxlade-Chamberlain a consultar na plataforma da STATSports as métricas medidas através do APEX Pod (fonte: STATSports)

Estes produtos foram testados por muitas das principais equipas, e os resultados demonstraram que o sistema da *STATSports* é o sistema mais preciso, confiável e consistente do mercado.

Relativamente aos produtos medidores de métricas, existem dois tipos para diferentes *targets: APEX Athlete Series* e *APEX Pro Series* (STATSports, 1981).

A. APEX Athlete Series

Este produto foi desenhado exclusivamente para os atletas. Este tipo de produto mede métricas como a distância percorrida, máxima velocidade, sprints, mapa de calor (áreas de maior foco

do jogador no campo) entre outros, perfazendo um total de 16 indicadores de desempenho. De entre os jogadores que utilizam este serviço, estão jogadores de renome mundial como Alex Oxlade-Chamberlain (Liverpool) e Raheem Sterling (Manchester City).

B. APEX Pro Series

Este medidor é projetado para desportos em equipa e tem a capacidade de calcular mais de 50 métricas em tempo real, correspondendo exatamente aos mesmos dados pós-jogo, sendo assim dados bastante precisos.

A diferença entre estes dois produtos é que o *APEX Athlete Series* é utilizado para ver apenas um jogador em específico, enquanto que o *APEX Pro Series* é para ver uma equipa de jogadores. Estes dois produtos envolvem tanto um *hardware* como um *software* diferente.

Para suportar um dos dois tipos de *Apex Pod*, existem dois tipos de produto:

- *Vest* – este produto é um colete que tem um bolso na parte de trás, com o objetivo de segurar o *Apex Pod* – aparelho medidor das métricas.
- *Base Layers* – camisola justa de manga comprida com um bolso na parte superior traseira para segurar o *Apex Pod*.

Na prática estes dois tipos de produto têm a mesma finalidade, que é segurar o aparelho que mede as métricas. No entanto, para desportos diferentes o colete pode ser mais adequado do que a camisola.

2.4. Plataformas desenvolvidas e usadas por académicos para recolha de dados

Uma das possíveis fontes de recolha de dados da plataforma são as aplicações de recolha de anotações de forma distribuída, dentro das quais gostaria de destacar três desenvolvidas em Portugal: a aplicação *diScout*, a *HiFootball* e a aplicação *uPATO*.

A aplicação *diScout*, do autor David Cadeirinhas, é um sistema direcionado para o processo de *scouting*, permitindo registar as ações de um determinado jogador durante uma partida. Os registos recolhidos pela aplicação são armazenados na base de dados e posteriormente originam dados estatísticos para os utilizadores consultarem. Este sistema permite aos clubes amadores e de formação melhorarem o departamento de recrutamento e também ter uma maior exposição dos seus jogadores (Cadeirinhas, 2019).

A aplicação *HiFootball*, do autor Bruno Barros, é uma aplicação desenvolvida no sistema operativo *Android* que ajudou a contribuir para a análise de jogos de futebol em tempo real. O principal objetivo é a recolha de anotações durante as partidas de futebol (como passes, remates,

livres, entre outros), sendo estas feitas pelos anotadores (desde técnicos a amantes de futebol). Esta aplicação permite anotar as interações dos jogadores nas partidas de futebol, tornando mais fácil a produção de relatórios detalhados para a análise do desempenho dos jogadores. A possibilidade de vários utilizadores utilizarem a aplicação para recolher anotações do mesmo jogo faz com que a informação possa ser comparada, fazendo assim com que a deteção de erros nas anotações seja possível e a informação seja mais credível e fidedigna (Barros, 2017).

Quanto à *uPATO (Ultimate Performance Analysis Tool)*, dos autores Fernando Martins, Filipe Clemente, Abel Gomes e Frutuoso Silva, é uma aplicação *Android* com o propósito de inovar na forma como a análise de jogo é feita em desportos coletivos, como é o caso do futebol. O próprio utilizador tem a possibilidade de escolher qual o desporto e quantos jogadores tem cada equipa (uPato, 2020). Esta aplicação permite não só recolher os dados da partida observada, como também codificar, importar, visualizar, analisar, calcular medidas e exportar dados (Clemente, 2019).

2.5. Impacto das redes sociais nos clubes e nos atletas

Hoje em dia o futebol é um mundo onde existem bastantes negócios e dinheiro envolvido, sendo que as redes sociais não ficam indiferentes a este processo. As redes sociais ajudam o negócio do futebol a crescer, criando oportunidades de patrocínios através de meios tão mediáticos como o *Twitter*, que fornece acesso ao conteúdo mais pessoal criado pelos próprios clubes ou jogadores. Esse tipo de conteúdo leva os clubes e jogadores a interagirem mais com os devidos interessados, aumentando assim a sua capacidade de influenciar as pessoas e a criar algo que os patrocinadores procuram (Pegoraro & Jinnah, 2012).

O desempenho das equipas continua a ser importante para o desenvolvimento dos clubes nas redes sociais, no entanto, o desenvolvimento e a implementação de uma estratégia de marca é benéfica para a maioria das equipas, ajudando-as a expandir nas redes sociais (Couvellaere & Richelieu, 2005). Por outro lado, segundo Vergeer e Mulder (2019), os jogadores não precisam de se esforçar nas redes sociais para obter seguidores pois este número está relacionado com o desempenho dos jogadores nos jogos e não com o número de publicações que fazem.

A importância das redes sociais no mundo do futebol deve-se também ao facto das publicações sobre desporto nas redes sociais terem alcançado paridade com as pesquisas feitas através dos meios tradicionais de comunicação em formato eletrónico e impresso (Pedersen, 2014).

Segundo um estudo de Browning e Sanderson (2016), os jovens estudantes e atletas usam o *Twitter* para manter o contato com outras pessoas, para comunicar com os seus seguidores e para ver comentários sobre as suas performances, tanto a meio do jogo como no final do jogo. Este fator é bastante importante pois pode influenciar o desempenho dos jogadores, e é devido a isso mesmo que certos clubes proíbem os seus jogadores de acederem a dispositivos móveis durante o intervalo. Importa dizer também que os resultados deste estudo demonstram que o *Twitter* é uma ferramenta de comunicação bastante importante, mas apresenta diversos desafios, como é o caso de ter de enfrentar comentários menos positivos e por vezes maldosos que podem mexer bastante com o psicológico dos jovens, tendo os clubes a responsabilidade de ter um departamento que ajude os jovens nesta vertente.

As redes sociais, como é o caso do *Twitter*, fazem com que a comunicação entre as pessoas seja feita de forma rápida e fácil, sendo hoje em dia ferramentas que permitem aos utilizadores terem bastante influência (Sanderson, 2011), por vezes até maior influência que os próprios clubes, e isso faz com que sejam criados aspetos positivos e negativos.

Para além de aspetos positivos como promover o clube, valorizar a marca ou gerar receita através da venda de *merchandising* do clube (Williams, Chinn, & Suleiman, 2014), um dos aspetos positivos para os clubes é poder comunicar com outras pessoas, fortalecendo o vínculo entre os adeptos e o clube, e poder partilhar informações sem ter um custo associado (Witkemper, Lim, & Waldburger, 2012), sendo diferente de ter um *site* próprio pois tem sempre custos associados. Outro aspeto positivo é o facto de o *Twitter* oferecer oportunidades para os clubes interagirem com os seus adeptos para melhorarem as suas experiências (Witkemper et al., 2012).

Para os jogadores poderá ser bastante positivo para ganhar reconhecimento e visibilidade, como também para melhorarem o seu desempenho em jogos futuros tendo em conta as críticas construtivas que são feitas (Browning & Sanderson, 2016), que devem ser levadas com seriedade.

Um dos aspetos negativos do *Twitter* nos jogadores das camadas jovens é o facto de estes não terem a capacidade que a maioria dos jogadores profissionais têm de lidar com comentários negativos. Um comentário pode levar jovens jogadores a produzir efeitos emocionais e psicológicos bastante negativos, o que depois pode levar a um impacto no seu desempenho (Browning & Sanderson, 2016). No entanto, para Gilberto Silva (Roseiro, 2020), o problema está no empenho do atleta e não nos comentários negativos, pois estes comentários devem ser um motivo para trabalhar mais e melhor.

Outro dos aspetos negativos é a possibilidade de um *tweet* inapropriado poder retirar a elegibilidade dos jovens jogadores. Enquanto que um *tweet* inapropriado por um jogador profissional pode ser punido pelo clube, mas não interfere no seu desempenho e maneira de jogar, se for feito por um jovem jogador das camadas jovens este pode perder a sua elegibilidade para jogar futebol ou ter punições severas por parte do clube (Browning & Sanderson, 2016).

Nas camadas jovens em Portugal, a importância do *Twitter* é enorme devido à falta de informação dos *media*. As redes sociais fornecem uma grande quantidade de dados que discutem os eventos desportivos gratuitamente, contrariamente aos serviços de anotação, sendo a agregação de *tweets* um recurso muito promissor para extrair diversos dados de jogos de futebol (Dijkshoorn, 2012).

Tanto os atletas como os próprios clubes utilizam bastante o *Twitter* para interagir com as pessoas e trazer conteúdo que não é colocado noutros locais. Um desses clubes, a título de exemplo, é o Clube Futebol Benfica⁹, mais conhecido por “Fofó”, que utiliza bastante esta rede social para divulgar o calendário de jogos das equipas, os resultados e diversas outras informações relativas às camadas jovens, sempre com uma *hashtag* associada (#FutebolBenfica e #SomosFofó) de forma à informação estar toda aglomerada.

A informação disponibilizada pelos clubes no *Twitter* é diversa e a maioria das vezes é publicada primeiro no *Twitter* do que nos *sites* dos clubes. Isto deve-se sobretudo ao volume de pessoas que visita esta rede social, sendo este bastante maior do que o volume de pessoas que visita os *sites* dos clubes. Segundo Price, Farrington e Hall (2013), o principal objetivo dos clubes terem *Twitter* é de cativar pessoas para acederem ao *website* oficial devido à quantidade de pessoas a que esta plataforma pode chegar.

2.6. Anotação e comentário coletivo: como é que o Zerozero coleta informação?

Sendo o Zerozero dos *sites* estudados (Tabela 1, Tabela 2 e Tabela 3), o *site* com mais informação relativamente às camadas jovens em Portugal, fica a faltar saber a razão pela qual têm mais informação.

A informação contida no Zerozero é produzida por quem trabalha no *site* e por colaboradores espontâneos (Figura 5). Para completar informações – sobre jogadores, sobre uma equipa, sobre o plantel de uma equipa, calendário de uma competição, resultados de jogos, sobre treinadores, sobre estádios, vídeos e fotografias – basta ser um utilizador registado no *site* e inscrever-se para colaboração. Uma vez registado no *site*, poderá sugerir alterações a um

⁹ <https://twitter.com/cfutebolbenfica>

jogador, como demonstra a Figura 6. Um colaborador tem também a possibilidade de administrar a informação de uma equipa ou controlar toda a competição (Zerozero, 2019).



O zerozero.pt cresce com a contribuição de todos os leitores.

COLABORADOR DA SEMANA

Veja agora como a sua ação pode ser importante no sentido de completar informação existente no zerozero.pt, e a forma como o poderá fazer.

NO ZEROZERO.PT PODERÁ...

- Completar informação sobre um jogador
- Completar informação sobre uma equipa
- Completar o plantel de uma equipa
- Enviar um calendário de uma competição
- Colocar resultados de jogos
- Completar informação sobre treinadores
- Completar informação sobre um estádio

Figura 5. Página de colaboração do Zerozero (fonte: Zerozero)

FICHA VIDEOS FOTOGRAFIAS PALMARÉS

Sugestão de dados José António

O zerozero.pt cresce com a contribuição de todos.

Apesar do rigor que tentamos assegurar nos conteúdos apresentados, se dispõe de alguma informação extra, agradecemos o seu interesse em melhorar o nosso serviço.

Apenas necessita de preencher os campos que pretende ver alterados.

Deixe-nos os seus dados de contacto:

Nome:

Email:

OBS.:

Se possível indique a fonte da informação no campo de observações

Marcar como jogador livre »

NOME COMPLETO:

NOME CURTO:

NASCIMENTO:

NACIONALIDADE:

DUPLA NACIONALIDADE:

NATURALIDADE:

PAÍS DE NASCIMENTO:

PÉ PREFERENCIAL:

POSIÇÃO DE RAIZ:

MAIS CONCRETAMENTE:

POSIÇÃO ALTERNATIVA:

MAIS CONCRETAMENTE:

ALTURA: Cm

PESO: Kgs

WEB:

FACEBOOK:

TWITTER:

YOUTUBE:

INSTAGRAM:

ESTADO DO JOGADOR:

AGENTE:

FOTOGRAFIA: Nenhum ficheiro selecionado

Tamanho máximo ficheiro: 400Kb
Formatos aceites: JPG, PNG e GIF

PARENTESCO:

Figura 6. Página para sugerir alterações aos dados de um jogador no Zerozero (fonte: Zerozero)

Com isto, tanto os clubes como o *Zerozero* ficam a ganhar, pois os clubes de menor dimensão ficam com dados estatísticos na *internet* e o *Zerozero* ganha conteúdo para o seu público.

2.7. Síntese da revisão da literatura

Neste capítulo esteve-se a estudar as diferentes fontes de informação que existem sobre futebol, tais como *sites* com informação desportiva, serviços e dispositivos de anotação de eventos e plataformas de recolha de eventos criadas por académicos, e foram feitas duas análises: uma sobre a relação das fontes de informação desportiva com assistências e classificações e outra sobre o impacto das redes sociais nos clubes e nos atletas.

Estas fontes de informação são muito diversificadas, quer em termos do público a que se destina (tais como clubes, jogadores, árbitros e adeptos), quer em termos dos meios que estão a ser utilizados (antigamente era exclusivamente em papel e hoje em dia tem vindo a dominar a informação no formato digital) sendo que mesmo os meios de comunicação tradicionais têm ambos os formatos, como é o caso dos jornais *A Bola*, *Record* e *O Jogo*.

Os meios digitais permitem uma publicação atempada e detalhada através de diferentes fontes institucionais (quer federação, associações de futebol ou os clubes) bem como outro tipo de fontes não oficiais, tais como o *Zerozero*. Nas fontes não oficiais também se destacam as que estão associadas a casas de apostas, pois esta associação está presente em grande parte das fontes estudadas.

No que respeita às camadas mais jovens, a informação está bastante ausente dos meios de comunicação mais tradicionais, no entanto, para estas camadas, as redes sociais são um meio muito relevante no contacto com os fãs e na promoção dos atletas.

Onde também se tem verificado uma enorme evolução para a produção destes conteúdos encontra-se nos sistemas de recolha de informação sobre os eventos, nomeadamente os serviços e dispositivos de anotação, tais como a *Opta* e a *STATSports*.

Feito este estudo das diferentes fontes de informação existentes no futebol, aquilo que parece importante e necessário como projeto, sobretudo nas camadas jovens (tendo em conta o impacto que este pode ter nos jovens e atletas), é a criação de uma plataforma que agrega não só a informação não oficial como a informação oficial.

3. Desenho e desenvolvimento do sistema

Neste capítulo são apresentados os requisitos, o desenho da arquitetura multicamada e todas as envolventes para o desenvolvimento do sistema.

Serão abordados os requisitos funcionais da plataforma, o Diagrama de Casos de Uso, as três camadas da arquitetura multicamada, as *web frameworks*, as bases de dados existentes, a implementação da plataforma, o desenho da base de dados, a forma de recolha de dados e a interface gráfica da plataforma, bem como a decisão no que toca às escolhas para o projeto.

3.1. Requisitos funcionais da plataforma

Os requisitos funcionais da plataforma são apresentados no Diagrama de Casos de Uso (Figura 7), estando descritos de forma mais detalhada no Anexo A.

3.1.1. Diagrama de Casos de Uso

A Figura 7 representa as possibilidades de ações que os utilizadores da plataforma podem tomar com base no que querem consultar ou inserir. Os utilizadores diferem apenas numa ação: “Inserir jogadores na base de dados”, que apenas é possível com um utilizador e uma palavra-passe, que estão alocados na coleção “*Users*” da base de dados. Os casos de uso (exceto os *includes*) estão associados aos requisitos do Anexo A.

A separação dos possíveis utilizadores justifica-se com a fiabilidade da fonte que insere os dados dos jogadores e com a utilização que cada utilizador poderá dar à plataforma. Os dirigentes são uma fonte oficial dos clubes de futebol e com isso fazem parte dos utilizadores registados, tendo a possibilidade de inserir dados dos jogadores dos próprios clubes. Relativamente aos olheiros, por estes terem um trabalho que requer um grande conhecimento de jogadores de futebol e com isso poderem ser uma possível fonte de dados, estão categorizados como utilizadores registados. Os atletas e os adeptos são categorizados como utilizadores não registados pois a plataforma para estes tipos de utilizadores serve apenas para consultar dados. Os dados inseridos na plataforma não são feitos por atletas e adeptos devido à possibilidade de adulteração dos dados dos jogadores.

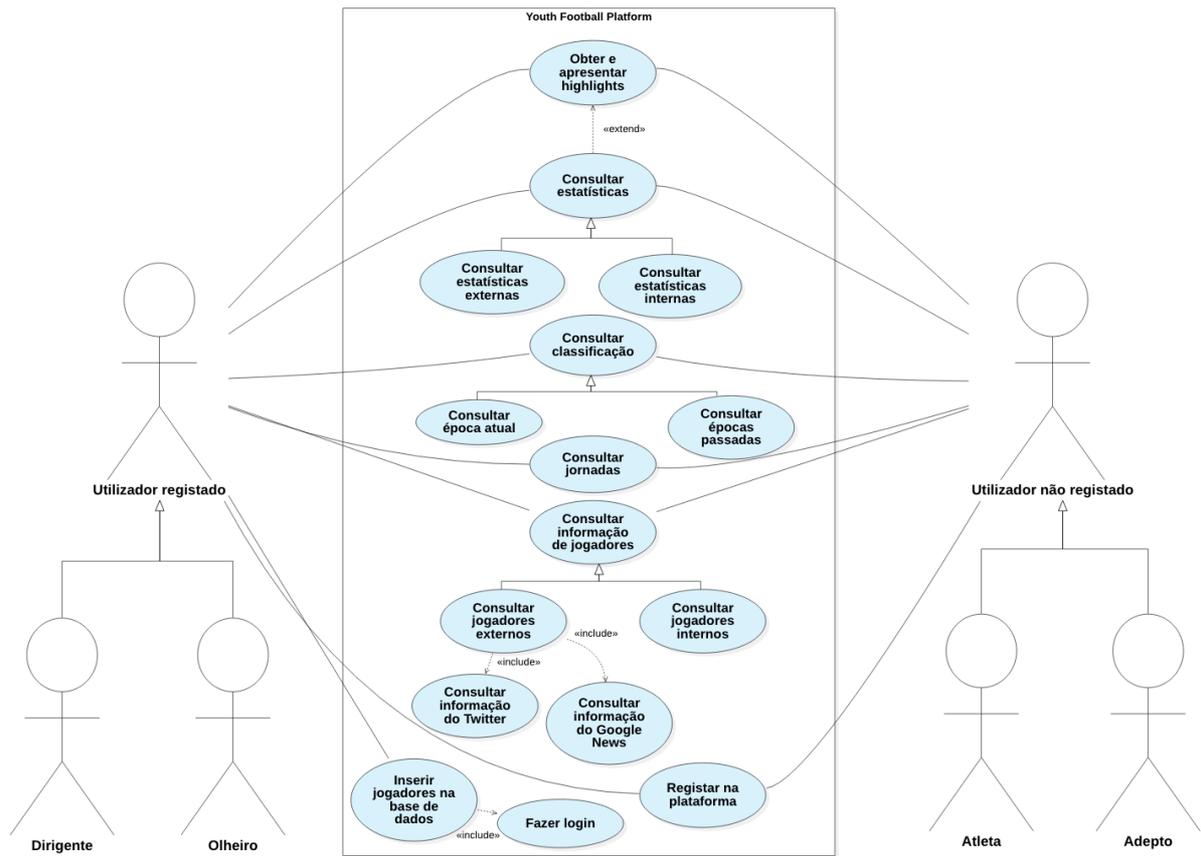


Figura 7. Diagrama de Casos de Uso

Do diagrama fazem parte dois atores e oito casos de uso, sendo todos importantes de detalhar:

- Atores:
 - Utilizador registado – representa o ator que tem todas as permissões possíveis na plataforma, tendo duas subcategorias: dirigente e olheiro;
 - Utilizador não registado – representa o ator que apenas pode consultar, não tendo a possibilidade de inserir jogadores na base de dados. Deste tipo de ator fazem parte o atleta e o adepto;
- Casos de Uso:
 - Obter e apresentar *highlights* – permite consultar as últimas notícias sobre as camadas jovens;
 - Consultar estatísticas – permite consultar as estatísticas de jogadores externas ou internas;
 - Consultar estatísticas externas – permite consultar estatísticas de jogadores extraídas diretamente do *Zerozero*;

- Consultar estatísticas internas – permite consultar estatísticas de jogadores extraídas da base de dados da plataforma;
- Consultar classificação – permite consultar a classificação das camadas jovens da época atual ou de épocas anteriores;
 - Consultar época atual – permite consultar a classificação das diversas categorias das camadas jovens de futebol em Portugal através da *FPF*;
 - Consultar épocas passadas – permite consultar a classificação de épocas passadas das diversas categorias das camadas jovens de futebol em Portugal através da *FPF*;
- Consultar jornadas – permite consultar as jornadas das diversas competições das camadas jovens de futebol em Portugal através da *FPF*;
- Consultar jogadores – permite consultar o perfil dos jogadores externos e internos;
 - Consultar jogadores externos – permite consultar o perfil dos jogadores, os últimos *tweets* relacionados, as últimas notícias relacionadas e os comentários dos especialistas no *Twitter*;
 - Consultar informação do *Twitter* – permite consultar os últimos *tweets* relacionados com o jogador, assim como os comentários dos especialistas no *Twitter*;
 - Consultar informação do *Google News* – permite consultar as últimas notícias relacionadas com o jogador;
 - Consultar jogadores internos – permite consultar o perfil dos jogadores que estão na base de dados da plataforma;
- Registrar na plataforma – permite ao utilizador registar-se na plataforma;
- Inserir jogadores na base de dados – permite inserir jogadores na base de dados da plataforma, tendo como requisito um utilizador e uma palavra-passe;
 - Fazer login – para inserir jogadores na base de dados da plataforma é necessário fazer login com um utilizador e palavra-passe;

De entre as várias funcionalidades da plataforma, há umas que são internas e outras externas. Isto deve-se ao facto de os dados “internos” da plataforma estarem alocados na base de dados *MongoDB*, enquanto que os dados “externos” são extraídos diretamente da base de dados do *Zerozero*.

Na secção 4.1 são feitos testes às ações descritas no Diagrama de Casos de Uso.

3.2. *Arquitetura multicamada*

A arquitetura usada neste sistema é uma arquitetura três camadas. Esta arquitetura é adotada pela maioria das aplicações *web* (Lee, Saha, Khan Yusufi, Park, & Karimi, 2009) e é composta por uma camada de apresentação, uma camada lógica e uma camada de dados, sendo que as três comunicam entre si, como é demonstrado na Figura 8. Cada camada realiza tarefas separadamente, podendo ser desenvolvida, mantida e alterada em qualquer momento (Tonar, 2013), fazendo assim com que não haja desorganização e processos complicados.

A camada de apresentação é responsável pela parte visual da plataforma, ou seja, é nesta camada que se constrói a interface que vai interagir com o utilizador (Lee et al., 2009). É nesta camada que é iniciado o processo da coleta de dados/informações, exibindo-os num formato legível para o utilizador (Tonar, 2013). Sendo a responsável pela parte visual, é aqui que são apresentadas as informações sobre as camadas jovens em Portugal, com a transformação dos dados já feita para que a sua exibição seja mais simples e intuitiva para o utilizador. A plataforma segue uma abordagem *RWD*, que faz com que as páginas da plataforma sejam adaptadas a qualquer dispositivo e tamanhos de janelas diferentes.

O papel desempenhado pela camada lógica é bastante importante, permitindo a coleta, transformação e distribuição de informações e dados em ambas as direções (Tonar, 2013). Esta camada também é responsável pela comunicação entre a camada de apresentação e a camada de dados, atuando como um conector entre as outras duas camadas (Lee et al., 2009). É na camada lógica que se encontra a subcamada de recolha de dados externos, de onde é importante salientar as técnicas usadas para obter esses dados: através da técnica de *web scraping* (ver subsecção 3.6.1), de *APIs* (ver subsecção 3.6.2) e através do *Selenium* (ver subsecção 3.6.3). De entre as diversas *web frameworks* existentes, a explicação para a escolha do *Django* está relatada na subsecção 3.3.3.

É na camada de dados que é feita a gestão do armazenamento interno dos dados relacionados com a plataforma e o seu acesso (Lee et al., 2009), tendo geralmente uma base de dados relacional ou não-relacional (Tonar, 2013), abordado mais à frente na secção 3.4, que também aborda a decisão de ter optado pelo *MongoDB*.

As fontes externas representam as fontes de onde são feitas as extrações de dados de diversos *sites* para que, no final de todo o processo, sejam exibidos na camada de apresentação. Estas fontes não estão dentro da camada de dados pois os dados exibidos são diretamente extraídos dos *sites* e não são guardados na base de dados da plataforma.

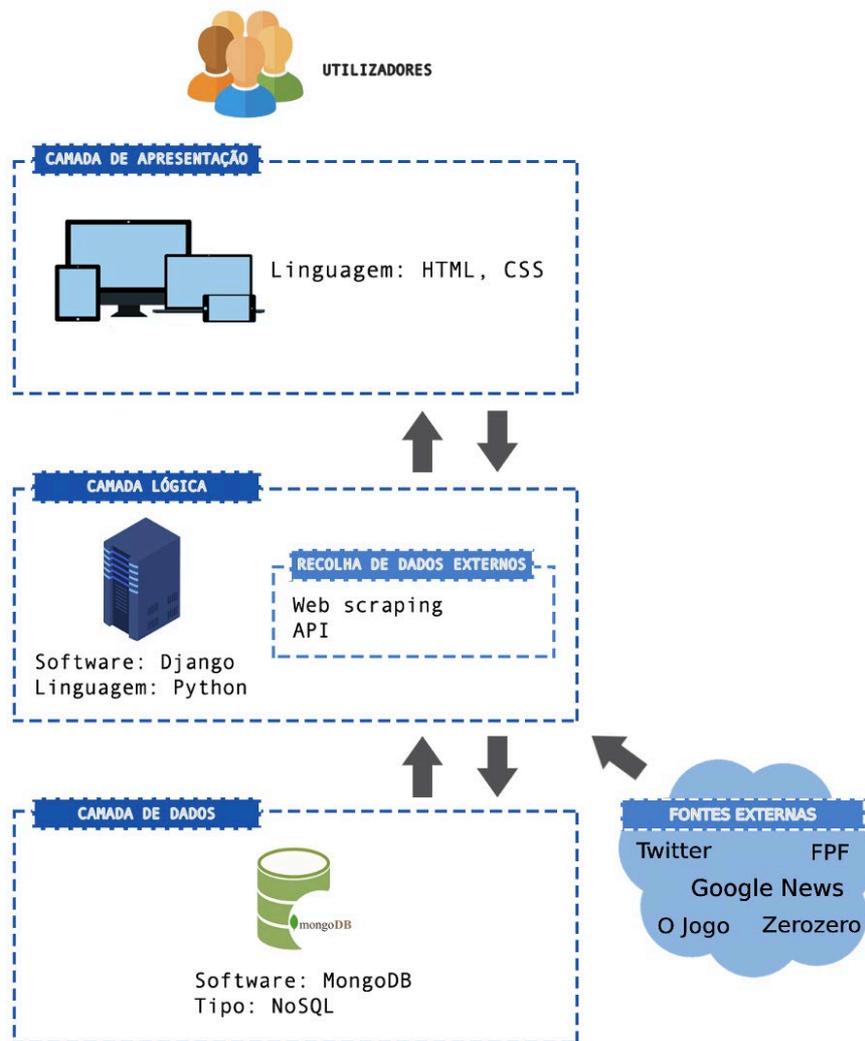


Figura 8. Arquitetura de 3 camadas: apresentação, lógica e dados.

Ao haver uma separação entre camadas, estas podem ser geridas independentemente, trazendo assim diversos benefícios como o tamanho das camadas, que pode ser gerido e adaptado conforme a sua necessidade; e o nível de segurança de cada camada, que também pode ser ajustado conforme o seu uso (Tonar, 2013).

O aspeto final da arquitetura do sistema, armazenamento e gestão do modelo de dados, é tratado pelo *software* de gestão da base de dados, que é acompanhado por um sistema para se comunicar com as aplicações (Lee et al., 2009), como o *MongoDB*, que é abordado na subsecção 3.4.2.1.

3.3. Camadas de apresentação e lógica: Web Frameworks

O desenvolvimento das camadas de apresentação e lógica é suportada por uma *Web Framework*.

Uma *Web Framework*, ou estrutura *web*, é uma coleção de bibliotecas ou módulos que permite aos programadores desenvolver aplicações *web* sem a necessidade de lidar com detalhes de baixo nível, como é o caso de protocolos, *sockets* ou *threads* (Python, 2020).

As *Web Frameworks* facilitam o modo de desenvolvimento de aplicações dos programadores, pois oferecem uma estrutura para estas mesmas. As estruturas automatizam a implementação de soluções comuns, reduzindo o tempo de desenvolvimento e permitindo que os programadores tenham mais tempo para a parte lógica das aplicações (Petlovana, 2020). Estas fornecem suporte para várias atividades, como interpretar pedidos (manipular *cookies* e sessões), produzir respostas (apresentar os dados em *HTML* ou em outros formatos) e armazenar dados. (Python, 2020)

Há diversas *frameworks* e alguns exemplos serão abordados nas próximas subsecções 3.3.1 e 3.3.2.

3.3.1. Full-Stack Frameworks: estado da arte

Uma *Full-Stack Framework*, ou estrutura de pilha completa, é uma solução completa com bibliotecas configuradas para funcionarem totalmente em conjunto. Suporta o desenvolvimento de serviços *back-end*, interfaces *front-end* e armazenamento de dados (Petlovana, 2020), o que corresponde às três camadas (camada de apresentação, lógica e de dados) da arquitetura multicamada da plataforma, descritas na secção 3.2.

3.3.1.1. Full-Stack Framework Django

O *Django*¹⁰ é uma *framework* de alto nível que incentiva o desenvolvimento rápido, com menos código e com um *design* limpo e eficiente. Permite criar aplicações de alto desempenho e tem o objetivo de automatizar o máximo possível, seguindo o princípio *DRY (Don't Repeat Yourself)* (Python, 2020), ou seja, reduzir a duplicação de código e os problemas que surgem dessa prática. Segundo o *site* JetBrains (2019), a *full-stack framework Django* foi a mais utilizada em 2018.

Esta ferramenta tenta incluir, por omissão, todos os recursos necessários em vez de oferecê-los como bibliotecas separadas. Alguns destes recursos, a título de exemplo, são a autenticação, o encaminhamento de *URL*, *ORM (Object-Relational Mapper)* e esquemas de migração de base de dados. Com estes recursos o *Django* é altamente escalável, rápido e versátil (Petlovana, 2020).

¹⁰ <https://www.djangoproject.com>

O *Django* tem muitas vantagens e destaca-se pelo seu rico ecossistema por conter muitas aplicações de terceiros, que podem, ou não, ser integradas no projeto mediante os requisitos do programador; pela maturidade, pois é uma ferramenta criada em 2005 e está em melhoria constante; pelo painel de administrador, que é gerado automaticamente e ajuda bastante na gestão da plataforma; pelo *SEO* (Search Engine Optimization), pois é possível gerar *URLs* e *links* legíveis de *sites* usando as palavras-chave mais relevantes e as melhores práticas de *SEO*; pela possibilidade de adicionar *plugins*, dando como exemplo o *Google Maps*¹¹; pelas bibliotecas, pois o *Django* permite que os programadores usem bibliotecas e assim automatizar processos; e pelo seu *ORM*, que ajuda na interação entre o *developer* e a base de dados (Zublenko, 2020).

3.3.1.2. Outras Full-Stack Frameworks

Para além do *Django*, existem outras ferramentas úteis e bastante populares entre os programadores de *Python*, como é o caso do *TurboGears* e *web2py* (Python, 2020).

A ferramenta *TurboGears* foi criada como uma camada *full-stack* e agora é uma *framework WSGI* (*Web Server Gateway Interface*) que pode atuar como *Full-Stack Framework* ou *Microframework* (Makai, 2020b). Usa bibliotecas do *Python*, mais propriamente *SQLAlchemy*, *Repose*, *Genshi* e *Tosca Widget* para a criação de aplicações orientadas a base de dados mais rapidamente, utiliza a técnica *AJAX* (*Asynchronous Javascript and XML*) para o servidor e navegador e *ORM* para a base de dados (Python, 2020).

O *web2py* foi inspirado no *Ruby on Rails* (web2py, 2020), uma *framework* que tem o foco no desenvolvimento mais rápido, fácil e com menos linhas de código (Ruby on Rails, 2020). Também foi inspirado no *Django*, diferindo deste mesmo por ser mais compacto, mais fácil de aprender e não contem arquivos de configuração no nível do projeto (web2py, 2020).

3.3.2. Microframeworks: estado da arte

As *microframeworks* são compostas pela maioria das funcionalidades de uma estrutura completa (Petlovana, 2020). As estruturas são pequenas, de fácil compreensão e fáceis de usar (Habib, 2016).

Este tipo de estrutura tem, normalmente, pouca ou nenhuma dependência de bibliotecas externas, o que leva a ter prós e contras:

¹¹ <https://www.google.pt/maps>

- Prós: a estrutura é leve e há pouca dependência para atualizar e observar *bugs* de segurança;
- Contras: a partir de uma certa altura, o programador terá que trabalhar mais sozinho ou aumentar a lista de dependências adicionando *plug-ins* (“Introduction to Flask”, 2008).

3.3.2.1. Microframework Flask

O *Flask* é uma microestrutura que fornece ferramentas, bibliotecas e tecnologias que permitem criar uma *web applications* (“Introduction to Flask”, 2008). Segundo o *site* JetBrains (2019), a *microframework* mais utilizada em 2018 foi o *Flask*.

Esta estrutura foi projetada para iniciar o desenvolvimento de aplicações *web* de forma rápida e fácil, com a capacidade e possibilidade de expandir para aplicações complexas (Pallets Projects, 2020). O *Flask* é baseado na biblioteca *WSGI Werkzeug* e na biblioteca de *Jinja2* (“Flask Documentation”, 2010).

O *Flask* oferece sugestões para o programador, mas não impõe nenhuma dependência ou *layout* para o projeto, estando do lado do programador a escolha das ferramentas e bibliotecas que deseja usar. Existem muitas extensões fornecidas pela comunidade que facilitam a adição de novas funcionalidades (Pallets Projects, 2020).

3.3.2.2. Outras Microframeworks

Para além do *Flask*, existem outras *microframeworks* úteis e bastante populares entre os programadores de *Python*, como é o caso do *CherryPy* e *Bottle* (Python, 2020).

O *CherryPy* permite aos programadores criar aplicações *web* da mesma forma que fariam com qualquer outro programa orientado a objetos em *Python*. Faz com que seja desenvolvido menos código em menos tempo, o que é bastante benéfico para o programador (CherryPy, 2020).

A *Bottle* é uma *microframework* que foi originalmente projetada para construir APIs (Petlovana, 2020) e não tem dependências para além da biblioteca padrão do *Python* (Marcel Hellkamp, 2020). No entanto esta *framework* tem uma limitação que pode ser um entrave para alguns programadores, já que foi desenvolvida para programas pequenos com um máximo de até 500 linhas de código (Petlovana, 2020).

3.3.3. Escolha de plataforma e critérios de avaliação

A decisão no que toca à escolha da *framework* a usar foi bastante ponderada e estudada. Depois de bastante pesquisa e leitura, a estrutura escolhida foi uma *Full-Stack Framework*, mais especificamente o *Django*. A Tabela 6 demonstra uma análise de algumas características do *Django* e do *Flask* que são descritas nesta subsecção.

Segundo a TechnoLush (2019), uma *Micro Framework* é a preferida para os programadores de aplicações mais pequenas e uma *Full-Stack Framework* é a preferida para programadores de aplicações maiores em que a maior parte do trabalho inicial já está feito por esta estrutura.

Como um projeto de mestrado requer tempo para o seu desenvolvimento, a curva de aprendizagem ser mais lenta no *Django* não influencia na decisão final, embora de um ponto de vista racional e devido à ferramenta *Flask* ser considerada a mais adequada para desenvolvimentos mais simples e mais rápidos (TechnoLush, 2019), a escolha poderia pender para a ferramenta *Flask*.

	Django	Flask
Recomendado para projetos pequenos		X
Recomendado para projetos grandes	X	
Curva de aprendizagem mais lenta	X	
Documentação mais completa	X	
Simplicidade		X

Tabela 6. Características Django vs. Flask

Muito embora estes critérios apontassem para a escolha *Flask*, a decisão recaiu sobre o *Django*: suportado pela imensa documentação e tutoriais do *site* oficial. Esta documentação torna o *Django* bastante intuitivo e mais fácil de aprender a trabalhar com a ferramenta.

Por último, embora não seja uma razão justificada tecnologicamente, a escolha sobre a *full-stack Framework Django* foi por esta ser mais rica e com maiores potencialidades futuras comparativamente a uma *Microframework* (TechnoLush, 2019), tornando-a mais desafiante do ponto de vista académico e pessoal. Adicionalmente, a opção pelo *Django* poderá potenciar a evolução deste protótipo para um sistema mais completo e sofisticado.

3.4. Camada de dados: tecnologias de base de dados

Uma base de dados é uma coleção organizada e estruturada de informações ou dados, e normalmente estão armazenados no sistema de um computador. Tipicamente uma base de dados é controlada por um *DBMS*, ou seja, um sistema de gestão de base de dados (Oracle,

2020a). Para Makai (2020a), uma base de dados é uma abstração do sistema de arquivos de um sistema operacional que facilita aos programadores a criação de plataformas e aplicações que criam, leem, atualizam e eliminam dados com muita frequência.

Nas bases de dados mais comuns, os dados são tipicamente modelados em linhas e colunas de tabelas para tornar o processamento e a consulta mais fáceis e eficientes (Oracle, 2020a).

Existem vários tipos de base de dados (Oracle, 2020a), das quais fazem parte a base de dados relacional (*SQL*) e a não-relacional (*NoSQL*).

3.4.1. Bases de dados relacionais – *SQL*: estado da arte

Este tipo de base de dados é a opção mais eficiente e flexível de aceder a informação estruturada (Oracle, 2020a), e também é o modelo de base de dados que lidera o ranking da DB-Engines (2020), com sete *DBMS* no *top 10*.

É baseado no modelo relacional, uma maneira intuitiva e direta de apresentar dados em tabelas (Oracle, 2020b), e a conexão entre estas mesmas é feita através de *foreign keys*, ou seja, chaves estrangeiras (Makai, 2020a). Estas tabelas contêm linhas e cada uma destas é um registo (DB-Engines, 2020b), que por sua vez contêm um *ID* que as identifica. Quanto às colunas de cada tabela, são mais concretamente atributos, que servem para facilitar o estabelecimento de relações entre dados (Oracle, 2020b) e assim gerar registos uniformes (DB-Engines, 2020b).

De forma a garantir que os dados estejam sempre claros e acessíveis, as bases de dados relacionais seguem determinadas regras de integridade. Um exemplo de uma regra de integridade é que linhas duplicadas não são permitidas numa tabela, e isso faz com que não haja informações incorretas na base de dados (Oracle, 2020b).

Segundo o *ranking* do DB-Engines (2020a), os *RDBMS* mais populares junto dos programadores são o *Oracle*¹², *MySQL*¹³ e o *Microsoft SQL Server*¹⁴.

3.4.2. Bases de dados não-relacionais – *NoSQL*: estado da arte

A base de dados não-relacional tem ganho bastante popularidade na última década, muito por causa dos dados gerados com *smartphones* e dispositivos *IOT*, que têm feito crescer a quantidade de dados produzidos (Schwarze, 2020). Com o aumento das aplicações de *Big Data* (MongoDB, 2020b), esta tornou-se uma ótima solução para armazenar, consultar e analisar um grande volume de dados (Schwarze, 2020).

¹² <https://www.oracle.com/>

¹³ <https://www.mysql.com>

¹⁴ <https://www.microsoft.com/pt-pt/sql-server/sql-server-2019>

São usadas para necessidades de *big data* e aplicações em tempo real que geram diversos tipos de dados, como imagens, vídeos, *posts* nas redes sociais e documentos (Schwarze, 2020).

Uma base de dados não-relacional, ou *NoSQL*, permite que os dados não estruturados e semiestruturados sejam armazenados e alterados ao contrário de uma base de dados relacional, que define logo como é que os dados inseridos na base de dados são compostos (Oracle, 2020a).

Este tipo de base de dados está, na maioria das vezes, subdividido em quatro categorias de armazenamento: documento, coluna, grafos e chave-valor (Schwarze, 2020).

O *NoSQL* com armazenamento de documentos é a categoria mais comum das quatro possíveis e geralmente são documentos *JSON* (Schwarze, 2020) ou documentos *XML* (Guru99, 2020b). Cada documento é um objeto que contém um código único com um conjunto de informações (Guru99, 2020b).

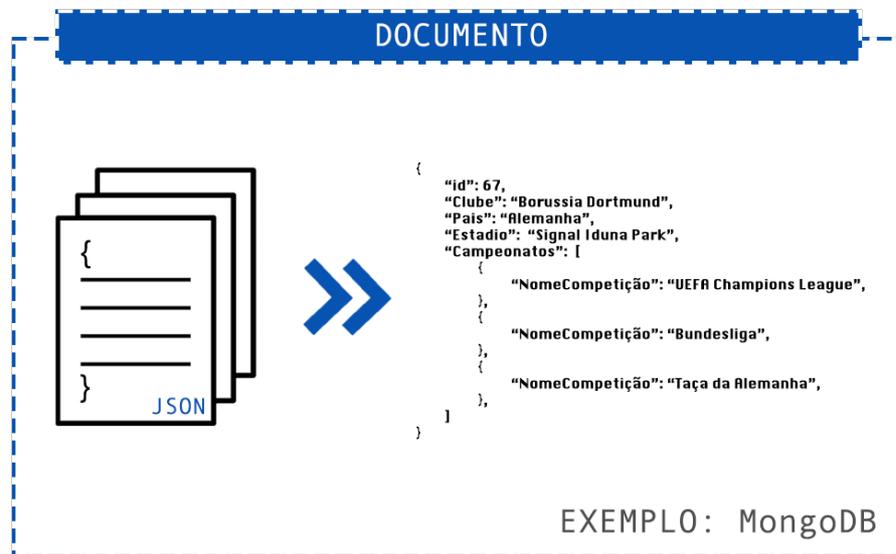


Figura 9. NoSQL orientado a documento

Armazenamento em coluna é usado para armazenar *Big Data* em vários conjuntos de dados. Este tipo de armazenamento é semelhante às bases de dados relacionais pois a sua estrutura suporta consultas em *SQL* (Schwarze, 2020).

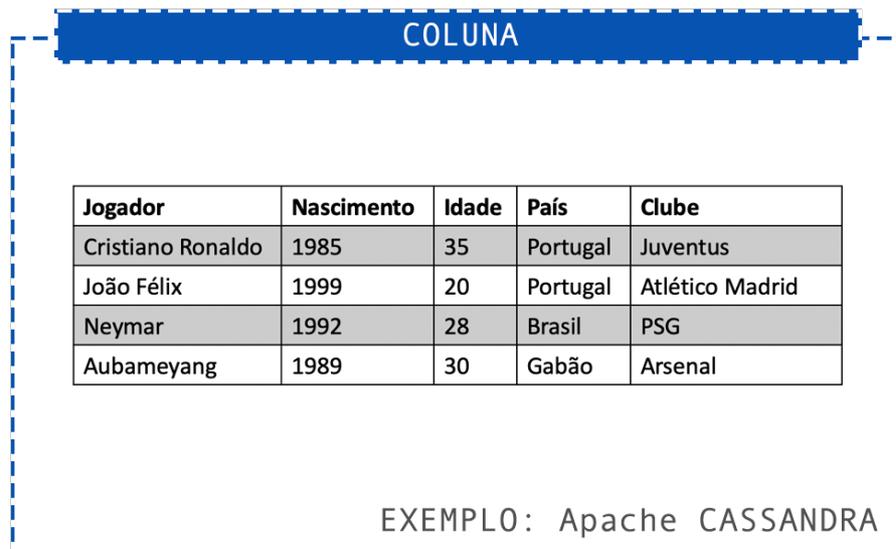


Figura 10. NoSQL orientado a coluna

Quanto ao *NoSQL* orientado a grafos, este faz a gestão de dados sobre sistemas em rede que podem crescer muito (Schwarze, 2020). Armazena entidades e as relações entre estas mesmas (Guru99, 2020b), sendo que está otimizado para a consulta de dados organizados em nós e arestas – ou seja, os conectores entre os nós (Schwarze, 2020). Este modelo é muito usado onde exijam dados fortemente ligados.

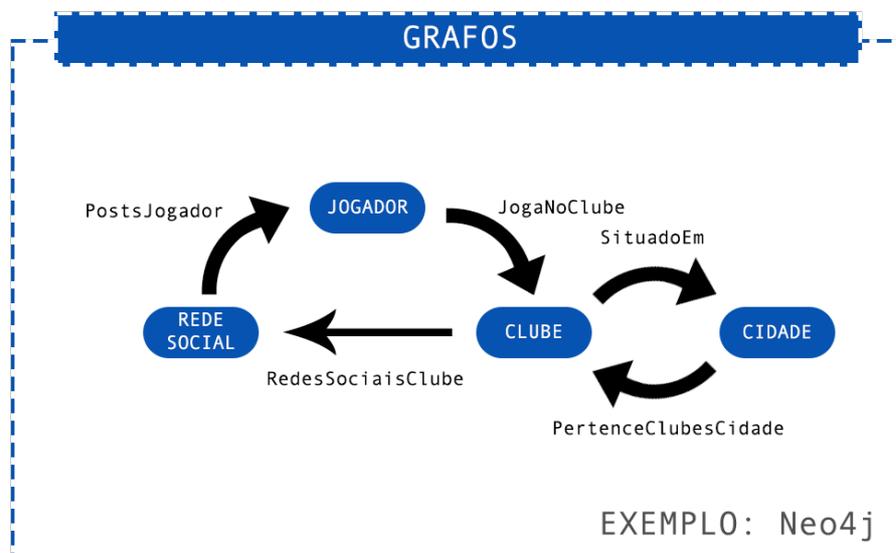


Figura 11. NoSQL orientado a grafos

O *NoSQL* chave-valor consiste no armazenamento de uma chave que é o identificador do valor associado (Schwarze, 2020), ou seja, os dados são armazenados em pares chave-valor (Guru99, 2020b). Se o valor associado à chave for um documento, este armazenamento poderá ser classificado como armazenamento de documentos (Schwarze, 2020). Este tipo de armazenamento foi feito para projetos de grande escala (Guru99, 2020b) com consultas

estruturadas para retornar rapidamente informações específicas para um utilizador, item ou entidade (Schwarze, 2020).

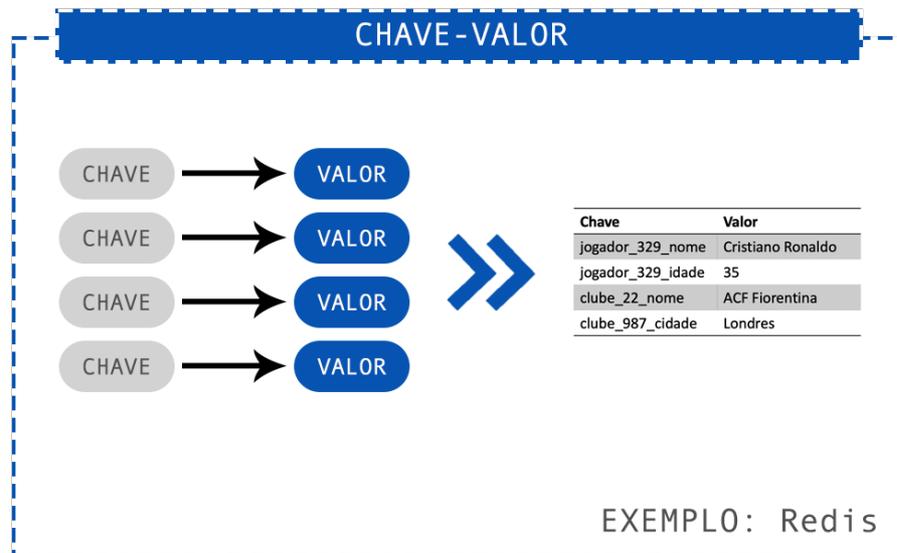


Figura 12. NoSQL orientado a chave-valor

As bases de dados *NoSQL* são mais escaláveis e flexíveis do que uma base de dados relacional que requer um esquema predefinido, oferecendo um design com um esquema flexível que facilita a atualização da base de dados para lidar com os requisitos de aplicações em mudança. Como a maioria dos dados gerados hoje em dia não são estruturados, estes encaixam-se melhor numa base de dados relacional do que num modelo estruturados em linhas e colunas (MongoDB, 2020b).

Segundo o ranking do DB-Engines (2020a), a base de dados *NoSQL* mais popular é o *MongoDB*.

3.4.2.1. Base de dados *MongoDB*

O *MongoDB* é uma base de dados *NoSQL* com armazenamento de documentos *JSON* (MongoDB, 2020a) e é orientada a objetos, simples, dinâmica e escalável (Jayaram, 2016).

Nasceu em 2009 e tem ganho bastante mercado, sendo a base de dados não-relacional que tem mais downloads feitos e mais desenvolvimentos realizados (MongoDB, 2020b). Foi coroado como líder em *DBMS* pelo Gartner¹⁵, líder mundial em consultoria e pesquisa, que abrange tanto bases de dados relacionais como não-relacionais (MongoDB, 2020b).

Nas bases de dados *MongoDB* os documentos são armazenados em coleções, onde são feitas operações de pesquisa (*queries*) e indexação (Politowski & Maran, 2014), tendo como

¹⁵ <https://www.gartner.com/>

principal objetivo implementar um armazenamento de dados que forneça alto desempenho, disponibilidade e escalamento automático (Jayaram, 2016).

O *MongoDB* utiliza o *BSON*, forma binária dos documentos *JSON*, para armazenar os documentos em coleções. Este formato utiliza “_id” como chave primária, tendo associado um valor exclusivo “ObjectId” que é automaticamente gerado pelo *MongoDB*. Uma das vantagens de usar *BSON* é que este mesmo permite indexar e mapear internamente as propriedades do documento (DataFlair, 2020).

Outro aspeto importante do *MongoDB* é o balanceamento de carga. Este permite que grandes volumes de dados que necessitam ser processados possam ter o seu tráfego distribuído por diversas máquinas, ajudando assim o utilizador a não ter o processamento dos dados interrompido. Caso algum nó ou máquina deixe de funcionar, este problema é resolvido pelo balanceamento de carga pois outros nós irão manter o processamento dos dados (DataFlair, 2020).

3.4.3. Escolha da base de dados e critérios de avaliação

A decisão no que toca à escolha do tipo de base de dados a usar recaiu sobre uma base de dados *NoSQL*. Este tipo de base de dados é facilmente escalável, não necessita de administradores na base de dados, é mais rápida, mais eficiente e evoluiu num ritmo bastante rápido (Ameya, Anil, & Dikshay, 2013).

Os dados são armazenados de várias formas, sendo bastante flexível, e assim faz com que não haja a obrigação de ter uma estrutura pré-definida, o que possibilita a inserção de dados vindos de diversas fontes, como é o caso do *Twitter* (ver subsecção 3.6.2.1) e da *FPF* (ver subsecção 3.6.1.1) que têm dados não estruturados. As base de dados *NoSQL* são escalonáveis horizontalmente, o que significa que o tráfego está mais fragmentado e assim pode tornar-se maior e mais poderoso, ótimo para conjuntos de dados de grande volume ou em constante mudança (Sharma, 2020).

Quanto à base de dados escolhida dentro das diversas opções *NoSQL*, a escolha foi o *MongoDB* por diversas razões, sendo uma delas a sua potencialidade e o enorme crescimento que teve entre os programadores desde o início da sua existência.

Segundo um estudo de Parker et al. (2013) em que compararam o *Microsoft SQL Server* com o *MongoDB*, o *MongoDB* obteve melhor desempenho no que toca a *inserts*, *updates* e *queries* simples. Esta é uma boa solução para conjuntos de dados maiores que estão em

constante mudança ou com *queries* menos complexas, tendo uma estrutura da base de dados menos rígida (Parker et al., 2013).

Abordando outro estudo feito por Politowski e Maran (2014) que compara a performance de uma base de dados relacional com uma não relacional, concluiu-se que o *MongoDB* obteve melhores resultados que o *PostgreSQL*¹⁶ no ambiente de testes proposto.

Para uma aplicação com uma grande carga de consultas à base de dados, o *MongoDB* é uma ótima alternativa (Politowski & Maran, 2014). Para o projeto em questão será utilizada a técnica de *web scraping*, inserindo-se totalmente de acordo com o paradigma do *MongoDB*.

3.5. Implementação da plataforma

A Figura 13 representa o Diagrama de Atividades da plataforma do ponto de vista do utilizador, que descreve os acontecimentos do sistema dependendo da ação do utilizador. Este diagrama é iniciado com a ação “Obter e apresentar *highlights*” que se refere à página inicial da plataforma. A partir desta página podemos ir diretamente para a página das estatísticas ou então exibir o menu para ir para outra página. As ações do menu “Obter e apresentar *highlights*”, “Exibir estatísticas”, “Exibir classificação”, “Exibir jornadas”, “Exibir jogadores” e “Exibir contactos” estão descritas com mais detalhe na Tabela 7. Após o utilizador exibir o que pretende ou inserir o jogador na base de dados, há a possibilidade de voltar a mostrar o menu para continuar a navegar na plataforma ou então terminar a atividade.

As principais opções do menu que o utilizador poderá tomar dentro da plataforma (Figura 13), como já foi referido anteriormente, estão pré-definidas e implementadas, sendo explicado todo o seu processo na Tabela 7. É nesta tabela que é apresentada a estratégia da base de dados e as possibilidades que o sistema pode tomar dependendo da situação em causa.

¹⁶ <https://www.postgresql.org>

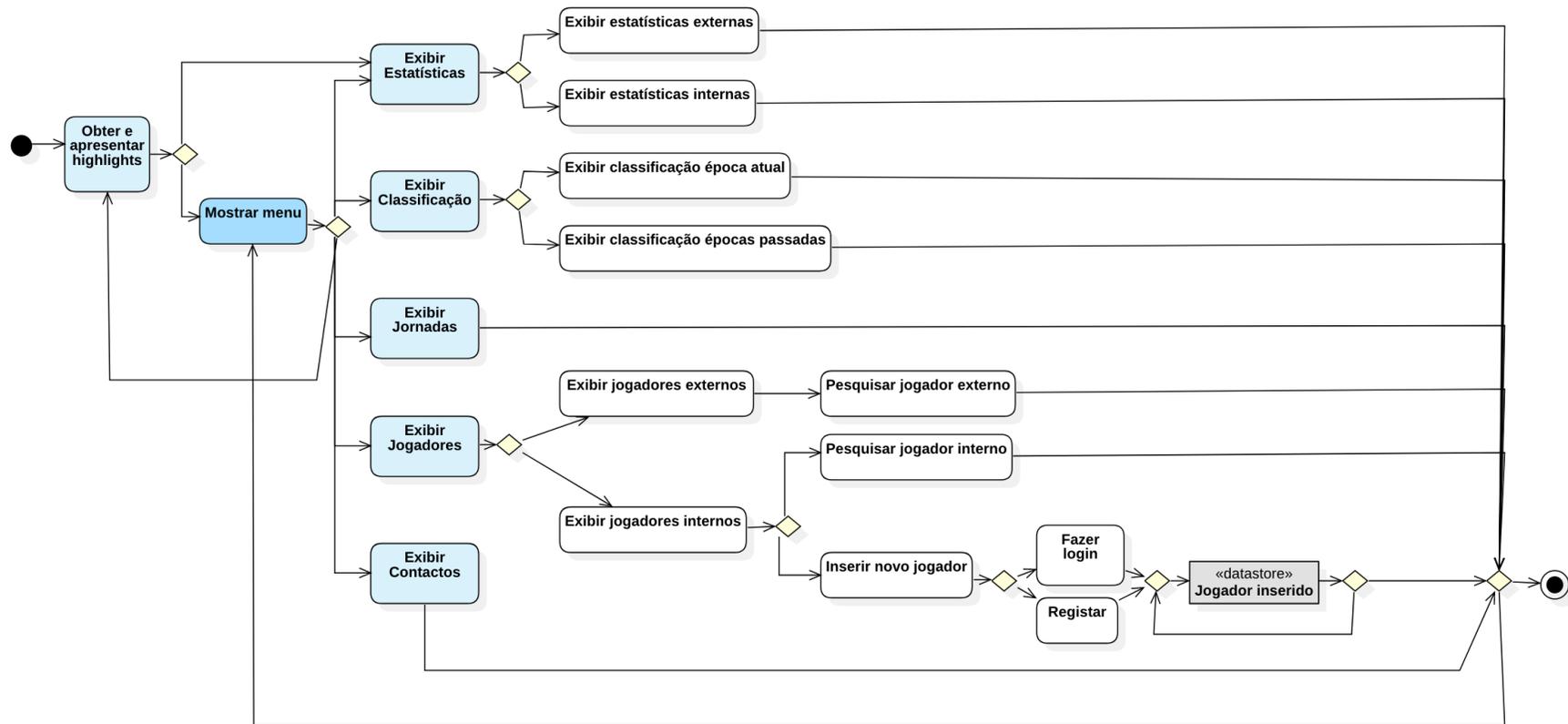


Figura 13. Diagrama de Atividades da plataforma do ponto de vista do utilizador

Página	Ação	Descrição	Parâmetros	Dados utilizados	
				Base de dados interna	Base de dados externa
<i>Home</i>	Obter e apresentar highlights	Mostra a página inicial da plataforma que contem as três notícias mais recentes das camadas jovens.	N/A	N/A	Dados extraídos através de <i>web scraping</i> . Ver secção 3.6. Fonte: <i>O Jogo</i>
Estatísticas	Exibir estatísticas	Mostra informação específica sobre um jogador e gráficos estatísticos associados.	É pedido um parâmetro de entrada com o nome do jogador.	Dados guardados referentes ao jogador que o utilizador procura.	Dados extraídos através de <i>web scraping/Selenium</i> . Ver secção 3.6. Fonte: <i>Zerozero</i>
Classificação	Exibir classificação	Mostra a classificação dos campeonatos das camadas jovens em Portugal.	É necessário escolher a época que pretende obter a classificação e escolher uma opção para cada uma das três <i>dropdowns lists</i> mostradas.	N/A	Classificação do campeonato escolhido através da técnica de <i>web scraping</i> . Ver secção 3.6. Fonte: <i>FPF</i>

Página	Ação	Descrição	Parâmetros	Dados utilizados	
				Base de dados interna	Base de dados externa
Jornadas	Exibir jornadas	Mostra as jornadas dos campeonatos das camadas jovens em Portugal.	É necessário escolher uma opção para cada uma das <i>dropdowns lists</i> mostradas.	N/A	Dados extraídos através de <i>web scraping</i> . Ver secção 3.6. Fonte: <i>FPF</i>
Jogadores	Exibir jogadores	Mostra informação específica sobre um jogador em específico e informação das redes sociais.	É pedido um parâmetro de entrada com o nome do jogador.	Dados guardados referentes ao jogador que o utilizador procura.	Dados extraídos através da técnica de <i>web scraping</i> e através de <i>APIs</i> . Ver secção 3.6. Fontes: <i>Zerozero, Twitter e Google</i>
Contactos	Exibir contactos	Mostra um formulário para enviar uma sugestão ou uma dúvida.	É pedido o nome, e-mail e mensagem.	N/A	N/A

Tabela 7. Lógica back-end da plataforma

3.5.1. Desenho da base de dados

A especificação e o desenho da base de dados da plataforma, assim como o seu modelo relacional estão demonstrados nesta subsecção.

As coleções criadas e utilizadas para armazenar os dados da plataforma foram as seguintes: “Users” e “Players”.

A primeira coleção “Users” guarda os utilizadores da plataforma, as respetivas palavras-passe e vários dados relativos ao utilizador, permitindo assim a estes utilizadores inserir os dados dos jogadores na base de dados da plataforma. A segunda coleção *Players* trata de guardar todos os dados relativos aos jogadores de futebol.

Estas coleções têm uma estrutura com determinados campos que são fundamentais para o seu propósito, como demonstram as seguintes tabelas Tabela 8 e Tabela 9:

Users
“_id”: str
“user”: str
“password”: str
“tipo_utilizador”: str
“nome”: str
“email”: str
“data_nascimento”: str
“genero”: str

Tabela 8. Coleção da BD – Users

Players
“_id”: str
“nome”: str
“nome_completo”: str
“nacionalidade”: str
“nascimento”: str
“clube”: str
“posicao”: str
“jogos”: int
“vitorias”: int
“empates”: int
“derrotas”: int
“golos”: int
“total_minutos”: int
“minutos_por_jogo”: int
“golos_por_jogo”: float
“autor”: str
“data_doc”: datetime

Tabela 9. Coleção da BD – Players

Devido às coleções terem bastantes campos diferentes, é essencial entrar em detalhe em campos que podem suscitar dúvidas.

No que toca ao primeiro campo de cada coleção, isto é, “_id”, estes são gerados automaticamente e servem para definir uma entrada específica na base de dados.

Em termos dos dados inseridos na coleção *Players*, há mais três campos que são gerados automaticamente: “minutos_por_jogo”, “golos_por_jogo” e “data_doc”. O campo “data_doc” (data do documento, ou seja, data da altura em que o registo é inserido na base de dados) é gerado automaticamente na altura de inserir os dados, enquanto que os outros dois campos são calculados consoante os dados inseridos nos campos “jogos”, “golos” e “total_minutos”.

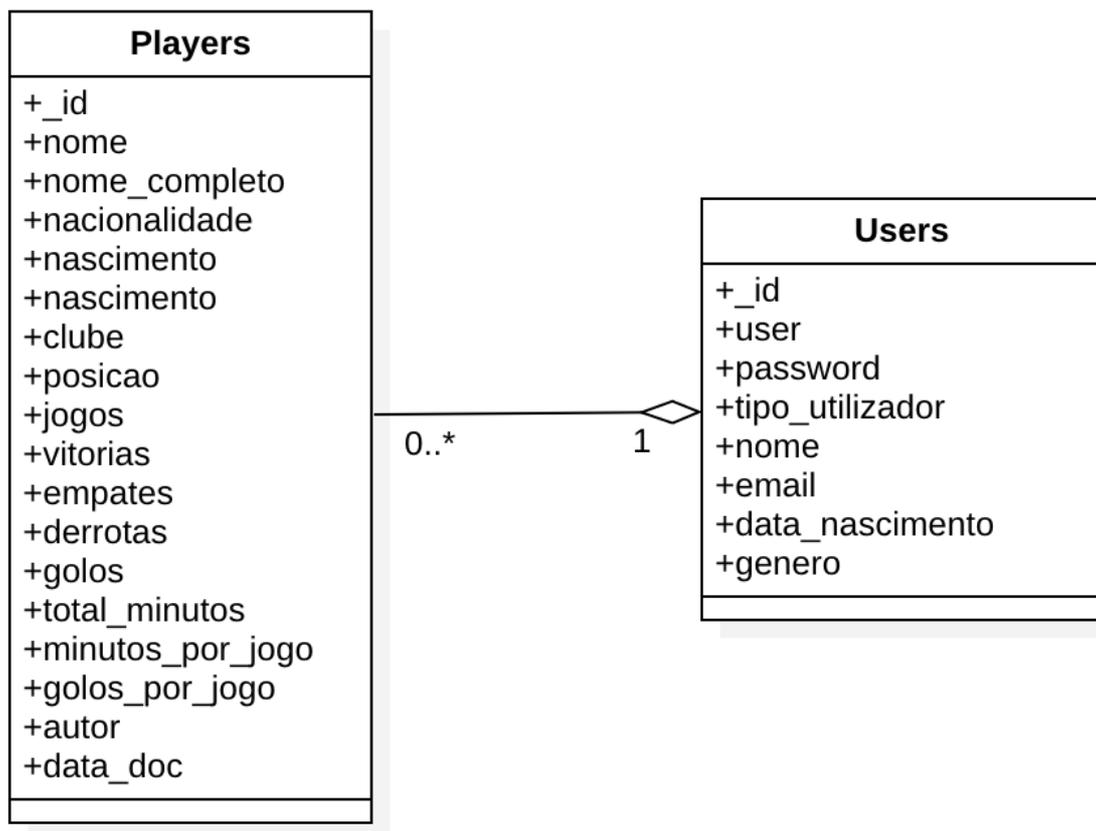


Figura 14. Modelo Relacional das coleções da base de dados

Como podemos verificar na Figura 14, existem um tipo de relação entre coleções na base de dados: relação de agregação.

A relação de agregação é feita quando as classes envolvidas são independentes uma da outra, ou seja, no caso de uma das classes deixar de existir, a outra continua a existir (Visual Paradigm, 2020). Esta relação é feita entre a classe *Users* e *Players* pois são independentes. Quanto à multiplicidade entre estas, é de um para muitos pois cada registo na classe *Players* pertence a um e um só utilizador (*Users*), enquanto que cada utilizador tem zero ou vários registos de jogadores (*Players*).

3.6. Recolha de dados

A recolha ou aquisição da informação é feita através de várias fontes (como podemos ver na Figura 15), dentro das quais *sites* institucionais, redes sociais, jornais desportivos, entre outros. Para recolher dados das redes sociais, Batrinca & Treleaven (2014) dão uma visão geral para quem procura coletar dados de redes sociais, através de *web scraping*, abordando *APIs* fornecidas por estas mesmas.

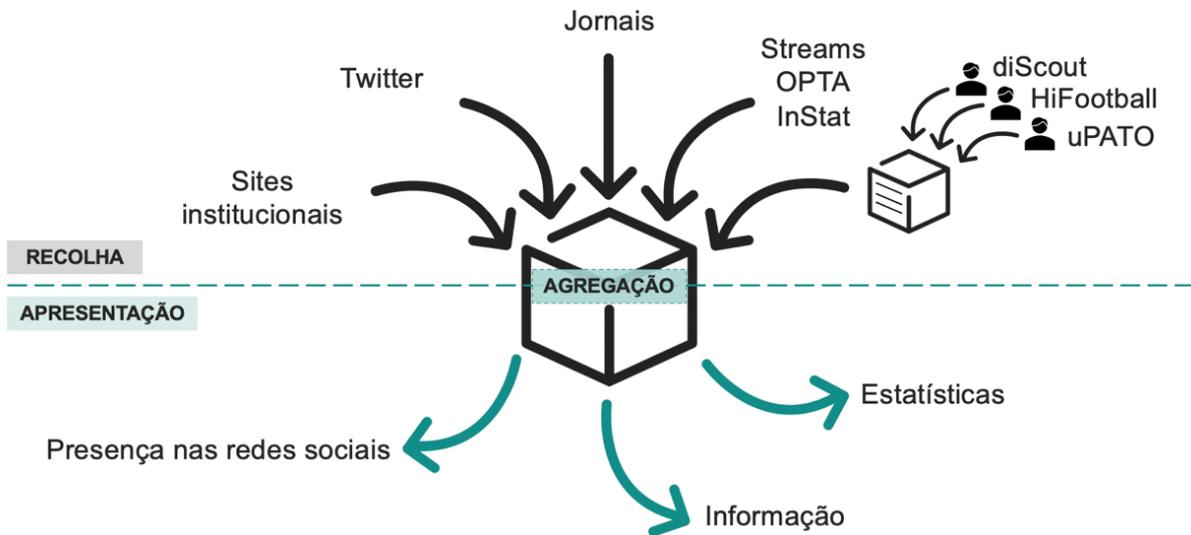


Figura 15. Esboço do processo de transformação da informação recolhida

Devido à pandemia *COVID-19*, os campeonatos relativos às camadas jovens foram interrompidos e, tanto a época 2019/20 como a época 2020/21, não representam uma época desportiva normal. Posto isso, os dados apresentados na plataforma são relativos à época 2018/19.

Nesta secção é apresentada a forma de obter dados e informações, de maneiras diferentes, através de *websites* com conteúdo relevante para a plataforma como é o caso da *FPF*, *Google*, *Zerozero*, jornal *O Jogo* e a rede social *Twitter*.

3.6.1. Recolha de dados através de *Web Scraping*

Web scraping é a prática de recolha de dados de um servidor *web* de forma automatizada. Para tal, é escrito em código um programa que consulta o código da página que pretendemos extrair informação, solicita os dados e analisa estes mesmos para extrair o que se pretende (Mitchell, 2018).

Antes de iniciar o desenvolvimento de um programa de automação de extração de dados, é bastante importante e necessária uma pesquisa sobre os termos de serviço do servidor da *web* que se quer extrair dados, pois muitos não permitem a técnica de *web scraping* (Breuss, 2019). Tipicamente, as páginas que não são permitidas aceder estão no ficheiro “*robots.txt*” da raiz de cada *site*, evitando assim a sobrecarga dos servidores do *site* através de solicitações ou até mesmo ocultar áreas confidenciais (como por exemplo as contas de utilizadores). A Figura 16 demonstra um exemplo de um ficheiro “*robots.txt*”, que é usado no *WhoScored*. As regras definidas neste ficheiro (Figura 16), como “*Disallow: /Predictions/*”, fazem com que o *site* bloqueie o acesso de rastreadores à pasta “*Predictions*”.

```
https://www.whoscored.com/robots.txt
User-agent: *
Disallow: /Accounts/
Disallow: /Predictions/
Disallow: /Users/
```

Figura 16. Ficheiro "robots.txt" do site WhoScored.com

Um dos desafios da técnica de *web scraping* é a variedade, pois todos os *websites* são diferentes e necessitam de uma abordagem própria. Tendem, no geral, a ter estruturas similares que facilitam o processo de *web scraping*, mas é sempre necessário um tratamento especial para cada um (Breuss, 2019). Um exemplo disso é a *tag* `<h1>` em *HTML*, que pode ter vários significados. Num *site* pode conter o título de um artigo, mas noutro pode significar o título do *site*, o que requer tratamento diferente para cada um dos casos (Mitchell, 2018).

Outro desafio é a dinâmica de alterações dos *websites*, pois estes estão em constante mudança e com isso a estrutura também pode mudar, o que é um problema. Para evitar este problema, é necessária uma manutenção constante do código que efetua o *web scraping* (Breuss, 2019).

3.6.1.1. Procedimento de recolha de dados do site da FPF

Para obter informações do *site* da *FPF* relativamente às camadas jovens em Portugal, foi necessária bastante investigação e tratamento dos dados. O *scraping* no *site* oficial da *FPF* serve para obter informações relativamente à classificação dos campeonatos e ao calendário de jogos das equipas.

O processo de extrair a informação da *FPF* é feito de duas formas: se a competição pertencer às “Competições FPF”, é apenas feito *web scraping* uma vez; se a competição for relativamente a uma Associação de Futebol (por exemplo, Viana do Castelo), é feito *web scraping* duas vezes. Isto deve-se ao facto de as competições da *FPF* serem as mesmas todos os anos, enquanto que os campeonatos dentro das AF mudam de ano para ano, fazendo com que não haja um padrão detetável nos *links*. Com isto, para extrair a classificação ou o calendário dos campeonatos de cada AF, é necessário fazer *web scraping* duas vezes, sendo uma delas para extrair os nomes e os *links* das competições e a outra para extrair a informação pretendida.

Como ponto de partida, o *URL* de cada AF está inserido numa estrutura, estando previamente definida no código da plataforma, como demonstra a Figura 17.

```
ASSOCIACAO_EPOCA_PT= {
  'af_algarve': 'https://resultados.fpf.pt/Competition/GetCompetitionsByAssociation?associationId=224&seasonId=',
  'af_angra': 'https://resultados.fpf.pt/Competition/GetCompetitionsByAssociation?associationId=216&seasonId=',
  'af_aveiro': 'https://resultados.fpf.pt/Competition/GetCompetitionsByAssociation?associationId=217&seasonId=',
```

Figura 17. Parte da estrutura da plataforma que guarda os links das AF da FPF

A título de exemplo, para obter automaticamente as ligas da Associação de Futebol de Lisboa associado a cada categoria (juniores, juvenis, entre outros) da época 2016/17, é utilizada a técnica de *web scraping* ao *link* <https://resultados.fpf.pt/Competition/GetCompetitionsByAssociation?associationId=229&seasonId=96>. Após a primeira extração de dados feita, é dada ao utilizador a opção de escolher a competição que pretende visualizar, através do qual é feito novamente *web scraping* para obter a classificação ou o calendário da competição. O último número do *link* - 96 - define a época que o utilizador pretende exibir (96 refere-se a 2016/17, 97 refere-se a 2017/18, 98 a 2018/19 e 99 é referente a 2019/20), enquanto que o número 229 é o *ID* da competição, que neste caso é a Associação de Futebol de Lisboa.

3.6.1.2. Procedimento de recolha de dados do site Zerozero

Para procurar os jogadores na página do *Zerozero* foi necessário investigar um padrão para conseguir fazer *scraping* à página, tal como foi feito com o *site* da *FPF*.

Inicialmente, depois de uma análise ao código da página inicial do *Zerozero*, foi descoberta uma página específica de pesquisa de jogadores - https://www.zerozero.pt/search_player.php. Através dessa página foram realizados diversos testes de pesquisa para descobrir um padrão na *query string* que, ao aplicar filtros manualmente, esta vai mudando consoante a pesquisa efetuada.

A título de exemplo, a intenção é procurar um jogador chamado Gonçalo Batalha. Assim sendo, o *URL* inicial é definido da seguinte forma: https://www.zerozero.pt/search_player.php?search_string=gonçalo+batalha&op=all. Para tornar esta pesquisa mais inteligente, são adicionados filtros específicos para o propósito da plataforma: jogador da modalidade futebol (*mod*=1), ordem da listagem na página (*ord*=*i*) sendo *i* de *increasing*, idade inferior a 18 anos (*ida*=1), género masculino (*fem*=0), de um clube português (*peq*=1) e no ativo (*sta*=0).

Posto isto, depois da análise feita com o exemplo dado, o *URL* final é: https://www.zerozero.pt/search_player.php?op=all&search_string=gonçalo+batalha&peq=1&fem=0&ida=1&mod=1&ord=i&sta=0&op=all.

Apesar de a forma de fazer *web scraping* ao *Zerozero* ser parecida com a da *FPF*, é necessário ter um certo cuidado na forma como o fazemos, isto porque o *Zerozero* tem um sistema de segurança no que toca aos acessos ao seu *website* para evitar o excesso de tráfego desnecessário. Por esta razão, a maneira de extrair dados do *Zerozero* levou a algumas alterações no meio do processo, como é explicado na subsecção 3.6.3 e no capítulo 4 - Testes e validações.

3.6.1.3. Procedimento de recolha de dados do site O Jogo

A técnica de *scraping* utilizada no jornal *O Jogo* foi com o intuito de ir buscar as três notícias mais recentes sobre camadas jovens.

Comparativamente com a técnica de *web scraping* utilizada tanto na *FPF* como no *Zerozero*, esta foi a menos complicada para a obtenção de dados. Foi feito inicialmente um pedido à página <https://www.ojogo.pt/futebol/formacao.html>, e depois, através de *BeautifulSoup*, obter as classes “t-g1-l1-m1” e “t-g1-l1-m2” que são referentes às notícias postas na página. Após isso é necessário segmentar o código obtido para receber as três notícias mais recentes.

3.6.2. Recolha de dados via API

Uma *API* (*Application Programming Interface*) permite aceder a dados de *websites* através de uma maneira predefinida (Breuss, 2019). Em termos teóricos tem a mesma finalidade do *web scraping*, que é obter os dados e informações de *websites*, mas na prática é feito de outra forma.

Com as *APIs* evita-se a análise do código do *website* que se pretende extrair dados, acedendo diretamente aos dados através dos formatos *JSON* e *XML* (Breuss, 2019).

Uma *API* diz-se predefinida pois lista várias operações que os programadores podem usar, com as respetivas descrições. Neste caso não é necessário saber o processo de como é feita a extração dos dados, o que facilita bastante. Economizar tempo aos programadores, menos linhas de código e maior consistência são algumas das vantagens a destacar (Hoffman, 2018).

Em comparação com a técnica de *web scraping*, as questões associadas à dinâmica de alterações na página *web* também se aplicam a uma *API*, no entanto o processo é mais estável por ser um processo pré-definido. Se a estrutura de um *site* for alterada, no caso das *APIs* não significa que a sua estrutura também é alterada (Breuss, 2019), sendo assim um pouco diferente da técnica de *web scraping*.

3.6.2.1. Procedimento de recolha de dados via API do Twitter

Para obter conteúdo do *Twitter* foi necessário utilizar uma das bibliotecas fornecidas pela comunidade desta rede social, a *TwitterSearch*¹⁷, que faz a ligação entre o código *Python* e a *API* do *Twitter*.

Utilizar esta biblioteca só é possível após a criação de uma conta de *developer* no *Twitter*, pois são necessárias quatro chaves únicas: *consumer key*, *consumer secret*, *access token* e *access token secret*. Isto permite um controlo por parte do *Twitter* à informação extraída pelos utilizadores das *APIs*. Importa dizer que cada *API* do *Twitter* tem um limite¹⁸ específico relativamente ao número de *tweets* que é possível extrair a cada período de tempo, sendo este limite definido por cada *API*.

Este método de recolha de dados é utilizado na plataforma para extrair informação relevante sobre os jogadores, não só dos perfis pessoais, mas também de *posts* de outras pessoas através de *hashtags*.

Na Figura 18 é exibido um exemplo da utilização desta *API*. Neste exemplo são exibidos *tweets* (*tweet['text']*) que contêm no texto “#Hashtag1” e “#Hashtag2”, juntamente com o nome de quem fez a publicação (*tweet['user']['screen_name']*).

```
from TwitterSearch import *

try:
    tso = TwitterSearchOrder()
    tso.set_keywords(['#Hashtag1', '#Hashtag2'])

    ts = TwitterSearch(
        consumer_key = 'aaabbb',
        consumer_secret = 'cccdde',
        access_token = '111222',
        access_token_secret = '333444'
    )

    for tweet in ts.search_tweets_iterable(tso):
        print('@%s tweeted: %s' % (tweet['user']['screen_name'], tweet['text']))

except TwitterSearchException as e: # take care of all those ugly errors if there are some
    print(e)
```

Figura 18. Exemplo de utilização da API *TwitterSearch* (fonte: <https://twittersearch.readthedocs.io/>)

A *API* do *Twitter* serve para dois propósitos: mostrar os últimos *tweets* relacionados com o nome do jogador e mostrar comentários dos especialistas da plataforma sobre o jogador exibido. Para um comentário aparecer na secção dos especialistas são necessários satisfazer dois requisitos: o utilizador do *Twitter* precisa de estar no código da plataforma, como demonstra a Figura 19; o comentário do especialista no *Twitter* necessita ter o nome do jogador em causa e

¹⁷ <https://developer.twitter.com/en/docs/developer-utilities/twitter-libraries>

¹⁸ <https://developer.twitter.com/en/docs/rate-limits>

duas *hashtags* (#YouthFootballPlatform e #Especialista). A criação e o destaque desta funcionalidade permitirão ter uma análise mais construtiva às exibições dos jogadores.

```
ESPECIALISTAS_TWITTER= {  
    'AndreRouiller': '@AndreRouiller'  
}
```

Figura 19. Excerto de código da plataforma

3.6.2.2. Procedimento de recolha de dados via API do Google News

A *API GoogleNews*¹⁹ é usada para pesquisar notícias que estão na secção de notícias do *Google*, estando atualmente na versão 1.4.1 (versão lançada dia 21 de agosto de 2020).

Esta *API* é menos robusta que a *API* utilizada para extrair dados do *Twitter*, pois para além de ter menos código para o intuito final de obter dados, ao contrário da *API* do *Twitter*, esta não necessita de chaves únicas para ser utilizada.

3.6.3. Recolha de dados via Selenium

O *Selenium*, desenvolvido por Jason Huggins em 2004, é uma estrutura automatizada de código aberto usada para validar aplicações *web* em diversos navegadores e dispositivos, mas também usada para a recolha de dados (Huggins, 2004).

Esta estrutura dispõe de diversas ferramentas como *Selenium IDE*, *Selenium Remote Control*, *Webdriver* e *Selenium Grid*, tendo cada uma delas um propósito diferente.

A ferramenta utilizada para a plataforma foi a *Webdriver*, que foi a primeira estrutura de teste *cross-platform* que controla o navegador *web* no nível do sistema operacional (Guru99, 2020c). Esta ferramenta faz uma simulação da interação do utilizador com as páginas *web* podendo ser uma alternativa ao *web scraping*, pois faz com que as páginas *web* com um forte sistema de segurança (para impossibilitar ataques de *hackers*) não detetem esta técnica para a recolha de dados.

Através desta ferramenta foi possível extrair os dados dos jogadores do *site Zerozero*, que tal como explicado anteriormente, tem um sistema de segurança rígido que não permite muitos acessos de forma simultânea. O procedimento efetuado é simples e grande parte do código desenvolvido anteriormente para fazer o tratamento dos dados do *web scraping* foi utilizado.

¹⁹ <https://pypi.org/project/GoogleNews/>

Na Figura 20 é exibido um exemplo da utilização do *Selenium*. Neste exemplo é feito um pedido ao site “*www.python.org*” através do *Google Chrome*²⁰, onde é lido o título do site através do comando “`print(driver.title)`”, é procurado o elemento “q” que corresponde à barra de pesquisas do site, é introduzido o texto e depois é obtido o *url* da pesquisa feita.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

driver = webdriver.Chrome('./chromedriver')
driver.get("https://www.python.org")
print(driver.title)
search_bar = driver.find_element_by_name("q")
search_bar.clear()
search_bar.send_keys("getting started with python")
search_bar.send_keys(Keys.RETURN)
print(driver.current_url)
driver.close()
```

Figura 20. Exemplo de utilização do Selenium (fonte: <https://www.browserstack.com/>)

3.7. Interface gráfica da plataforma

Nesta secção são descritos e mostrados os ecrãs possíveis de aceder através da *Youth Football Platform*, a plataforma desenvolvida.

O tipo de dispositivo escolhido e utilizado da Figura 21 à Figura 36, à exceção da Figura 22, para descrever e demonstrar os ecrãs da plataforma é *mobile*.



Figura 21. Página inicial da plataforma (versão móvel)

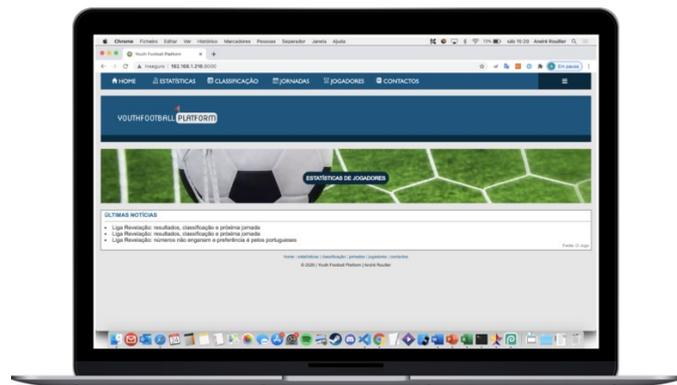


Figura 22. Página inicial da plataforma (versão desktop)

²⁰ <https://www.google.com/intl/pt-PT/chrome/>

A Figura 21 e a Figura 22 é o rosto inicial da plataforma, onde é destacada a página das estatísticas e são demonstradas as três últimas notícias do jornal *O Jogo* relativas às camadas jovens de futebol em Portugal, através da técnica de *web scraping*.



Figura 23. Página das estatísticas



Figura 24. Página das estatísticas externas

A Figura 23 representa a página das estatísticas, que se subdivide em duas: estatísticas externas e estatísticas internas. As estatísticas externas referem-se à informação extraída diretamente do *Zerozero* e estatísticas internas referem-se à informação extraída da base de dados da plataforma.

A Figura 24 exhibe a página de pesquisa das estatísticas externas, sendo exibidos também dois *tops* referentes às estatísticas externas e às estatísticas internas: jogadores mais procurados das estatísticas externas e melhores marcadores das estatísticas internas.

O ecrã das estatísticas internas é idêntico ao ecrã das estatísticas externas (Figura 24), daí não ser necessário a sua exibição.

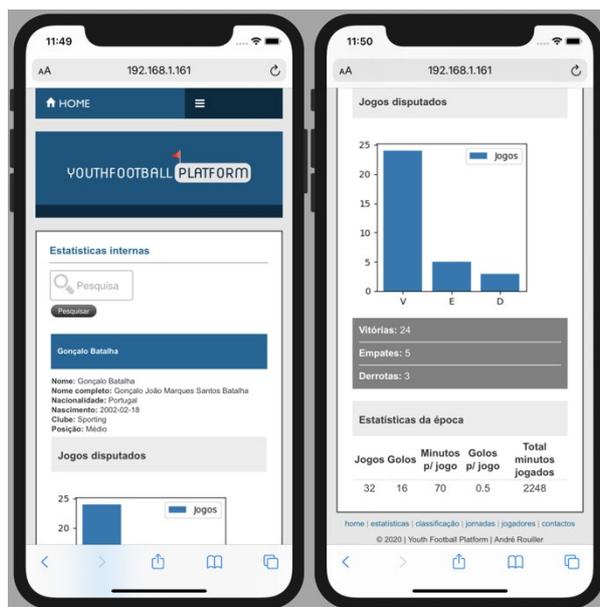


Figura 25. Página das estatísticas internas do jogador Gonçalo Batalha

A Figura 25 demonstra a visualização das estatísticas de um jogador, alocadas na base de dados da plataforma. Estas estatísticas incluem um gráfico com o número de vitórias, empates e derrotas, assim como os jogos, golos, total de minutos jogados e dois cálculos feitos com base nos dados anteriores: minutos por jogo e golos por jogo.



Figura 26. Página da classificação

A Figura 26 representa a página inicial da classificação que é dividida em duas subpáginas: classificação da época atual e classificação de épocas passadas. A classificação atual, como já foi referido anteriormente na secção 3.6, é relativa à época 2018/19, enquanto que a classificação de épocas passadas é referente às épocas 2016/17, 2017/18 e 2018/19.



Figura 27. Página da classificação da época atual

FASE FINAL									
POS	Equipes	JG	V	E	D	GM	GS	PTS	
1	Sl Benfica Sad	10	9	1	0	29	6	28	
2	Fc Porto Sad	10	6	0	4	14	15	18	
3	Sporting Cp. Sad	10	5	2	3	21	14	17	
4	Sc Braga Sad	10	3	4	3	15	18	13	
5	Vitória Sc. Sad	10	1	2	7	13	30	5	
6	CF Os Estremozenses	10	1	1	8	10	19	4	

Figura 28. Página da classificação do campeonato nacional de juvenis

A Figura 27 e Figura 28 são referentes à classificação atual das camadas jovens de futebol em Portugal. A Figura 27 é a exibição inicial da página, que após a escolha da associação de futebol e da categoria das camadas jovens, é despoletado outro *dropdown* com as competições possíveis dentro das possibilidades. Após este ecrã intermédio são mostradas as tabelas classificativas (Figura 28), podendo ser mais do que uma, dependendo se a competição tem mais do que uma fase.



Figura 29. Página da classificação das épocas anteriores

A página da Figura 29 é semelhante à página da Figura 27, no entanto relativa à classificação das épocas anteriores. Esta página tem mais um *dropdown*, que diz a época que o utilizador quer exibir.



Figura 30. Página das jornadas



Figura 31. Página da jornada 9 da 1ª divisão do campeonato nacional de juniores

A lógica visual das jornadas é bastante parecida à da classificação, mudando apenas na página intermédia, tendo esta mais um *dropdown* com as jornadas, como demonstra a Figura

30. A Figura 31 é o resultado final da consulta à jornada 9 da 1ª divisão do campeonato nacional de juniores.



Figura 32. Página inicial dos jogadores

A página inicial dos jogadores (Figura 32) é semelhante à página das estatísticas (Figura 23), com duas opções possíveis: jogadores externos e jogadores internos. Tal como nas estatísticas.

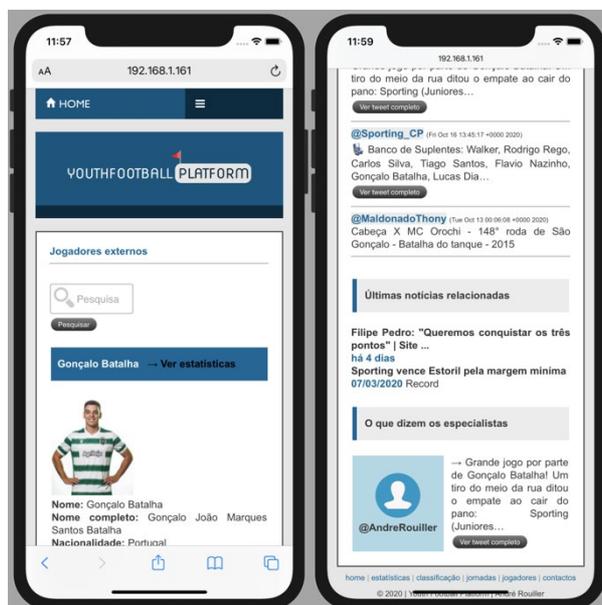


Figura 33. Página do jogador Gonçalo Batalha

A página de jogadores externos, como demonstra a Figura 33, exibe dados de diversas formas: dados extraídos do *Zerozero* através de *web scraping*, dados extraídos de uma *API* do *Twitter* e também dados extraídos de uma *API* do *Google*. A técnica de *web scraping* é utilizada para extrair o perfil do jogador, a *API* do *Google* serve para extrair as últimas notícias relacionadas com o jogador procurado e a *API* do *Twitter* serve para dois propósitos: mostrar os últimos *tweets* relacionados com o nome do jogador e mostrar comentários dos especialistas da plataforma sobre o jogador exibido. Importa dizer também que esta página tem um atalho ao lado do nome do jogador para ver as estatísticas do jogador exibido.



Figura 34. Página de jogadores internos

Na Figura 34 é exibida a página de pesquisa dos jogadores internos, havendo também a opção de inserir um novo jogador na base de dados. Para tal, será necessário inserir um utilizador e uma palavra-passe, como demonstra a Figura 35, ou registar-se na plataforma (Figura 36), onde o utilizador terá que fornecer alguns dados: utilizador, palavra-passe, tipo de utilizador, nome, *e-mail*, data de nascimento e o género.



Figura 35. Página de autenticação para inserir um jogador na base de dados

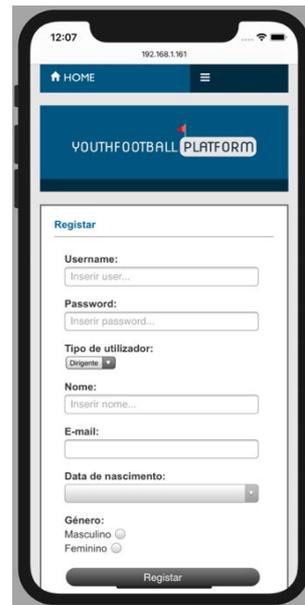


Figura 36. Página de registo na plataforma

Depois de efetuar o login, serão necessários diversos dados sobre o jogador em concreto que queremos inserir, como demonstra a Figura 37. Este formulário é convertido automaticamente num documento *JSON* (Figura 38) para ser inserido com sucesso na base de dados. Após isso já é possível encontrar o jogador através da barra de pesquisa demonstrada na Figura 34.



Figura 37. Página para inserir um jogador na base de dados da plataforma

```

1  {
2    '_id': ObjectId('5f71c3b9c5e9054d9055da02'),
3    'nome': 'Gonçalo Ramos',
4    'nome_completo': 'Gonçalo Matias Ramos',
5    'nacionalidade': 'Portugal',
6    'nascimento': '2001-06-20',
7    'clube': 'Benfica',
8    'posicao': 'Avançado',
9    'jogos': '44',
10   'vitorias': '31',
11   'empates': '8',
12   'derrotas': '5',
13   'golos': '20',
14   'minutos_por_jogo': '73',
15   'golos_por_jogo': '0.5',
16   'total_minutos': '3209',
17   'autor': 'André Rouiller',
18   'data_doc': '28/09/2020 - 11:06:33'
  }

```

Figura 38. Documento JSON do jogador Gonçalo Ramos

4. Testes e validações

Neste capítulo são apresentados os testes e validações realizados à plataforma desenvolvida, como também a justificação para os resultados obtidos.

Nos testes funcionais são feitos três tipos de testes: teste de requisitos, teste de caixa-preta e teste automatizado. Já nos testes não-funcionais é feito o teste de caixa-preta, que é apoiado por vários gráficos com os tempos de performance da plataforma, assim como a devida explicação para os valores obtidos.

De forma a conseguir separar as contribuições no atraso na plataforma e na rede, nos tempos obtidos nos testes efetuados, a máquina que serviu de servidor não foi a mesma onde foram realizados os testes.

Os requisitos funcionais utilizados para o teste de requisitos da plataforma estão descritos no Anexo A e os *links* para os vídeos dos testes realizados estão no Anexo B.

4.1. Testes funcionais

Na primeira fase deste capítulo são feitos os chamados testes funcionais. Os testes funcionais são realizados antes dos testes não funcionais e são baseados nos requisitos do utilizador (Anexo A). Ajuda a avaliar o comportamento da plataforma e é um tipo de teste que verifica se as funcionalidades estão em conformidade com a sua descrição. São comparados os resultados reais com os resultados esperados, podendo ser feito de forma manual ou automática (Guru99, 2020a).

O primeiro teste foi realizado com base no Diagrama de Casos de Uso (ver subsecção 3.1.1) e nos respetivos requisitos funcionais da plataforma. Ao percorrer os Casos de Uso é verificado se os requisitos funcionais estão a ser cumpridos. Este teste (Tabela 26) foi realizado com o intuito de demonstrar uma execução normal da plataforma, tendo sido testadas todas as funcionalidades possíveis, ou seja: obter e apresentar *highlights*, consultar estatísticas (tanto externas como internas), consultar classificação (época atual e épocas passadas), consultar jornadas, consultar jogadores (externos e internos), consultar informação do *Twitter*, consultar informação do *Google News*, fazer login, registar na plataforma e inserir jogadores na base de dados.

Para o segundo teste foi feito o chamado “teste de caixa-preta” (Tabela 27). Para além dos testes normais de funcionamento da plataforma, este tipo de teste avalia o *output* da plataforma mediante os dados inseridos não-comuns, ou seja, mediante *inputs* inesperados. Foram feitos testes às páginas “Estatísticas externas”, “Estatísticas internas”, “Jogadores externos” e

“Jogadores internos” em forma de comparação. Dos dez *inputs* diferentes, nas “Estatísticas externas”, oito *outputs* retornaram o esperado: nenhum jogador correspondente à pesquisa feita. No entanto, dois dos *outputs* retornaram o mesmo jogador (minutos 01:05 e 01:14) e sem razão aparente, até porque os dados desse jogador não contêm os *inputs* da procura. As “Estatísticas internas” obtiveram os dez *outputs* esperados, não retornando nenhum jogador com os *inputs* inseridos. A página dos “Jogadores externos” obteve o mesmo resultado que a página “Estatísticas externas” no que toca à extração de dados do *Zerozero*, pois o processo de extração feito em ambos os casos é idêntico. Por último, a página de “Jogadores internos” teve o mesmo resultado que as “Estatísticas internas” visto que a forma de pesquisa na base de dados é feita da mesma forma.

Um dos problemas enfrentados ao realizar os testes foi o *web scraping* ao site *Zerozero*. A maneira pensada para extrair os dados do *Zerozero* era utilizando a biblioteca *BeautifulSoup* através de pedidos. Ao correr apenas uma vez a página “Estatísticas externas”, eram feitos quatro pedidos. Ao tentar fazer uma nova pesquisa, o *Zerozero* despoletava o serviço *reCAPTCHA*²¹ (Figura 39) que ficava ativo durante dezenas de minutos e impedia a extração de dados.

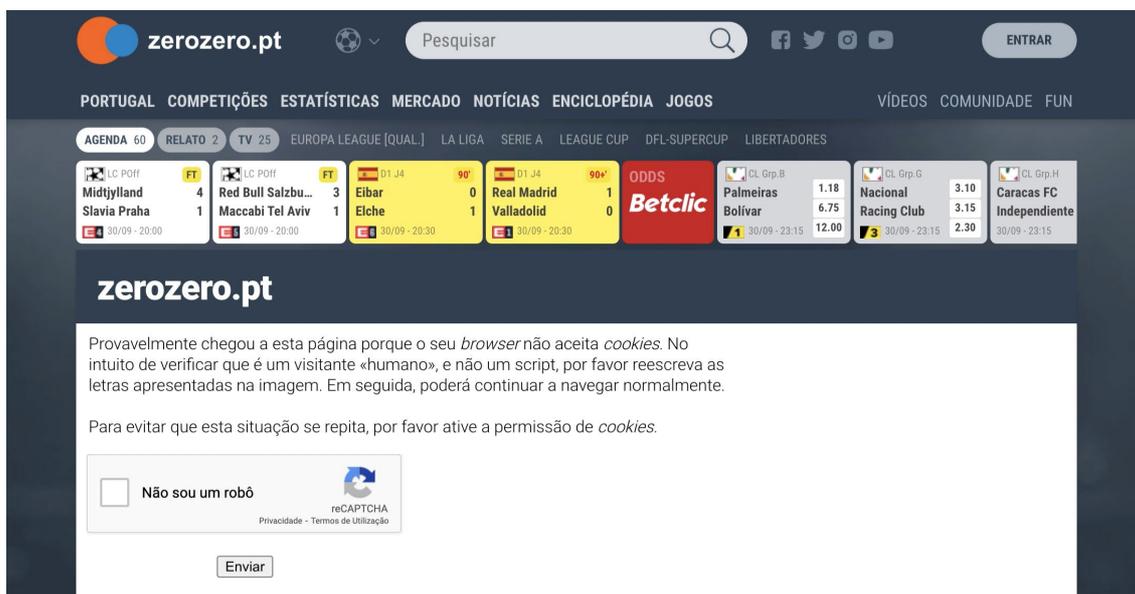


Figura 39. *reCAPTCHA* Zerozero

Como forma de amenizar este problema, a decisão partiu por deixar de fazer *web scraping* dessa forma e desenhar uma nova abordagem, utilizando o *Selenium*, descrito na subsecção 3.6.3. Importa dizer que parte do código-fonte utilizado para o *web scraping* foi aproveitado

²¹ <https://www.google.com/recaptcha/about/>

para a nova abordagem a partir do *Selenium*, o que foi bastante útil pois não foi necessário fazer um novo processo de raiz.

O terceiro e último teste funcional foi feito com base no *Selenium*, através da ferramenta *Jupyter Notebook*²², e consistiu em fazer 50 acessos seguidos automatizados às estatísticas externas (Figura 45), à classificação da época atual (Figura 47) e épocas passadas (Figura 48) e também aos jogadores externos (Figura 46), sendo que a ordem dos 50 jogadores utilizados para as estatísticas externas e jogadores externos está demonstrada na Figura 44. Este teste também contribuiu para vários resultados demonstrados na secção 4.2. O resultado do teste às estatísticas externas (Tabela 28) mostrou a totalidade dos 50 jogadores procurados. Como o processo de extração foi melhorado com o decorrer do projeto, como está descrito na subsecção 3.6.3, permitiu que o teste final fosse realizado com sucesso em todos os jogadores procurados. Os resultados dos testes automatizados à classificação da época atual (Tabela 29), épocas passadas (Tabela 30) e jogadores externos (Tabela 31), sendo os primeiros dois resultados da extração de dados da *FPF* e o terceiro o resultado das extrações de várias fontes, mostraram dados nos 50 pedidos de exibição efetuados, no entanto, na extração aos jogadores externos, a partir da 38ª extração (minuto 06:28), a *API* do *Twitter* excedeu o limite de *tweets* referido anteriormente (ver subsecção 3.6.2.1).

4.2. Testes não funcionais

A plataforma é ainda avaliada através de testes não funcionais, ou seja, testes que verificam aspetos não funcionais como o desempenho e a usabilidade da plataforma (Guru99, 2020a). Ao contrário dos testes funcionais, os testes não funcionais preocupam-se com o código-fonte da plataforma, pois a sua performance e usabilidade são dependentes do código escrito.

A validação da plataforma é feita com base no chamado “teste de caixa-branca”. Este teste baseia-se numa técnica de teste de *software* que avalia a estrutura interna da plataforma, ou seja, o código-fonte. Para tal, os gráficos desta secção ajudam a analisar os tempos de performance do código-fonte. Os tempos obtidos e mostrados nos gráficos são uma média de 50 resultados diferentes para cada teste efetuado.

Como podemos ver no gráfico da Figura 40, foram calculados os tempos médios para satisfazer os requisitos funcionais da plataforma (ver exemplos nas tabelas 32, 33, 34 e 35 no Anexo B). O tempo medido é calculado desde que o servidor *YFP* recebe o pedido do utilizador

²² <https://jupyter.org/>

até que envie de volta uma mensagem *HTTP 200*, ou seja, é o tempo que demora todo o processo dentro do tracejado da Figura 41.

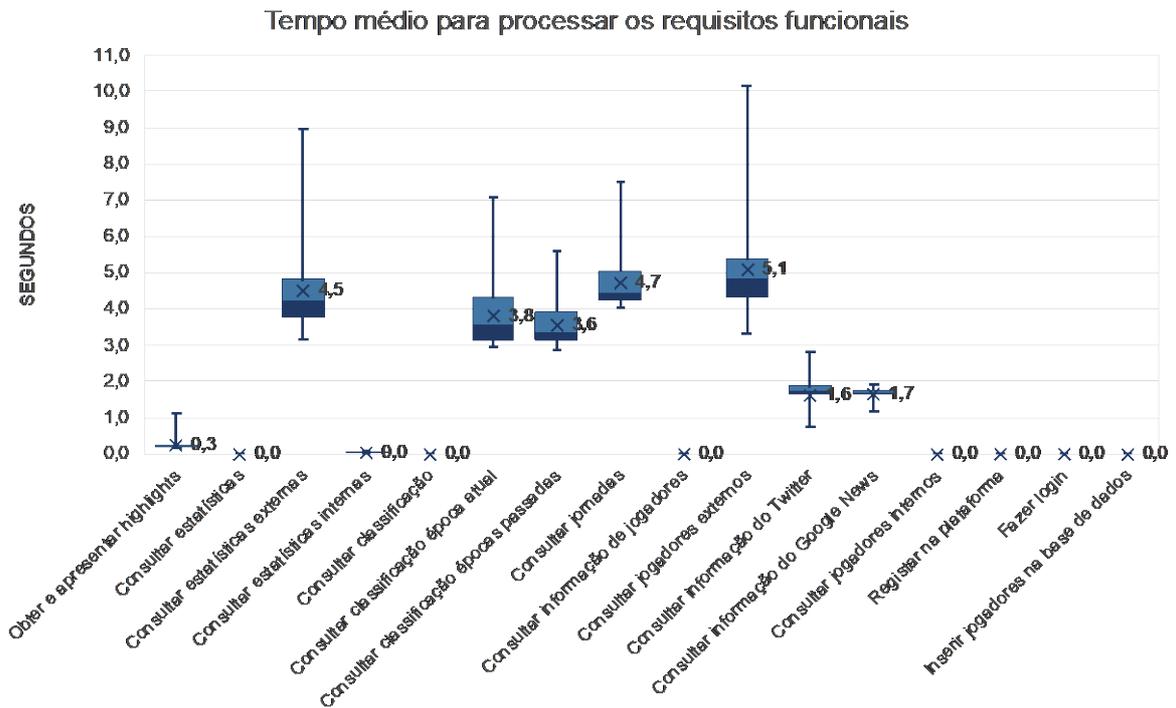


Figura 40. Tempo médio para processar os requisitos funcionais

Os testes automatizados efetuados para os testes funcionais também ajudaram a testar a parte não funcional da plataforma. Estes testes são feitos por um utilizador da plataforma (lado esquerdo da Figura 41), onde são enviados pedidos para o servidor da plataforma. A partir do momento que o pedido chega ao servidor, este começa a contar o tempo que demora a extrair os dados das diversas fontes (são feitos novos pedidos, mas desta vez entre o servidor e as fontes externas) até que envia de volta uma resposta ao utilizador. O código destes testes automatizados está no Anexo C.

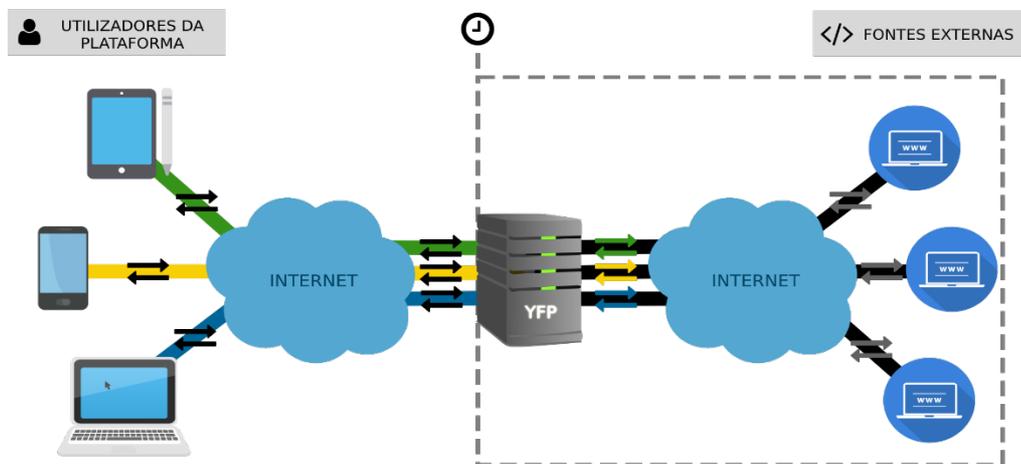


Figura 41. Interações com elementos externos da plataforma

Através da Figura 40 podemos concluir que os requisitos que demoram mais a ser exibidos são “Consultar jogadores externos”, “Consultar jornadas” e “Consultar estatísticas externas”, sendo estes falados com maior detalhe mais à frente. Isto deve-se ao facto de estes requisitos necessitarem de extrair dados de fontes externas, como demonstrado na Figura 41. Comparando com requisitos como “Consultar estatísticas internas” e “Consultar jogadores internos” que extraem dados da própria base de dados, podemos concluir que há uma diferença de 4,5 a 5,1 segundos entre a consulta de dados internos e externos. Importa realçar que de todos os requisitos que têm pelo menos uma extração de dados externos no seu processo, “Obter e apresentar *highlights*” foi a que obteve o menor tempo médio registado no agregado de 50 resultados diferentes: 0,3 segundos.

Das várias fontes de extração de dados, a *FPF* é utilizada para três requisitos diferentes: “Consultar classificação época atual”, “Consultar classificação épocas passadas” e “Consultar jornadas”, sendo importante comparar os tempos médios obtidos e demonstrados na Figura 40. A diferença de cerca de um segundo a mais entre a consulta das jornadas comparativamente com a consulta das classificações prende-se no facto de o processo de extração de dados para as jornadas necessitar de três *web scrapings* (um para exibir os nomes das competições da AF escolhida, outro para extrair o *link* da jornada específica e outro para extrair os dados que serão exibidos), enquanto que as classificações precisam de dois *web scrapings* (um para exibir os nomes das competições da AF escolhida e outro para extrair os dados que serão exibidos).

Comparando o tempo médio para exibir as estatísticas externas e estatísticas internas da Figura 40, é possível observar que o tempo médio para processar as estatísticas internas é quase nulo, enquanto que para processar a extração de dados externos leva, em média, 4,5 segundos. De forma a ser mais clara a razão para as estatísticas externas demorarem cerca de 4,5 segundos, o gráfico da Figura 42 mostra a fragmentação dos tempos para cada parte do código desta funcionalidade da plataforma. A extração dos dados iniciais do jogador ao *Zerozero* demora cerca de 1,7 segundos e o tempo necessário para criar o gráfico estatístico é quase nulo pois leva cerca de 0,1 segundos para ser gerado. O processo que demora mais tempo é a extração dos dados estatísticos, pois é feito em cerca de 2,7 segundos. Isto deve-se ao facto de ser o processo mais complexo, pois os dados utilizados encontram-se numa página do *Zerozero* que não é acessível através de um *link* genérico, mas sim através de uma simulação real no *Selenium*.

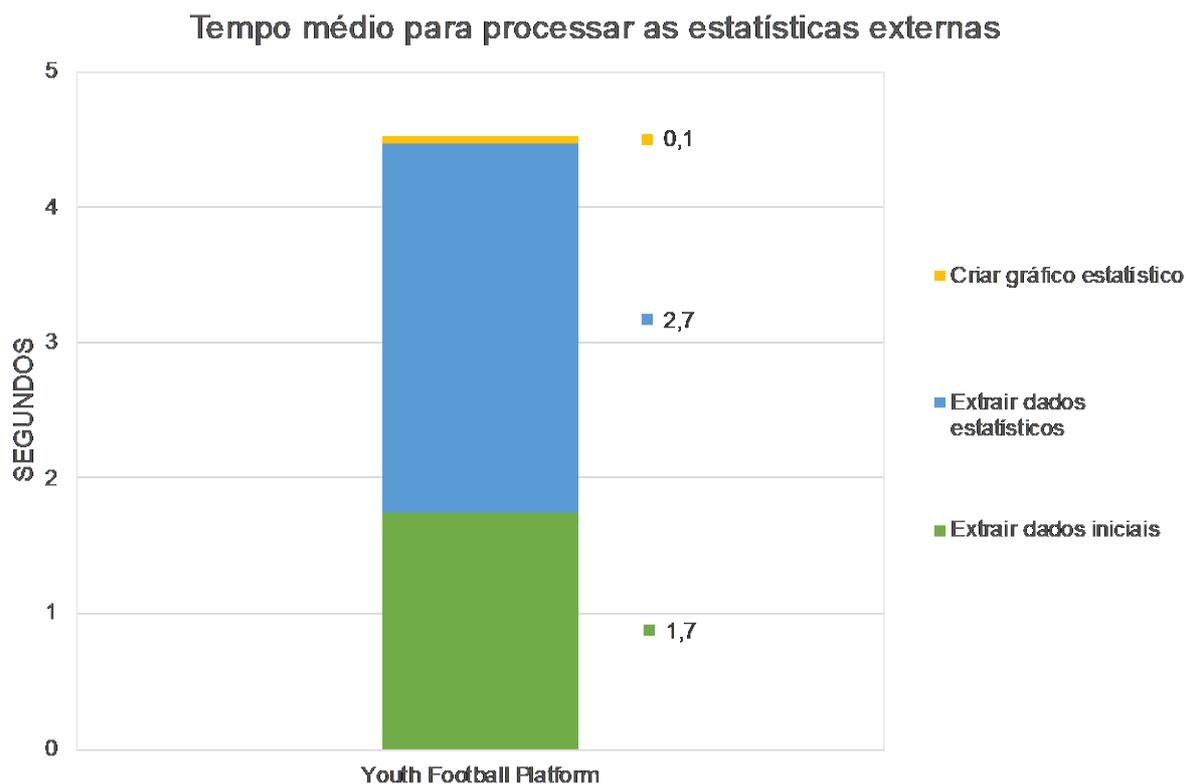


Figura 42. Tempo médio para processar as estatísticas externas

O último gráfico (Figura 43) com valores retirados dos testes efetuados à plataforma mostra a fragmentação do tempo médio total (5,1 segundos) pelos diversos processos dentro desta funcionalidade dos jogadores externos. O processo mais demorado é a extração de dados iniciais que demora cerca de 1,8 segundos para obter os dados iniciais do jogador procurado. Os processos “Extrair informação do *Twitter*” e “Extrair comentário do especialista” têm os mesmos tempos médios, sendo perfeitamente justificável pois ambos utilizam a mesma *API* para obter os dados pretendidos. Por último, a extração de informação do *Google News*, feita através de uma *API*, leva aproximadamente, 1,7 segundos.

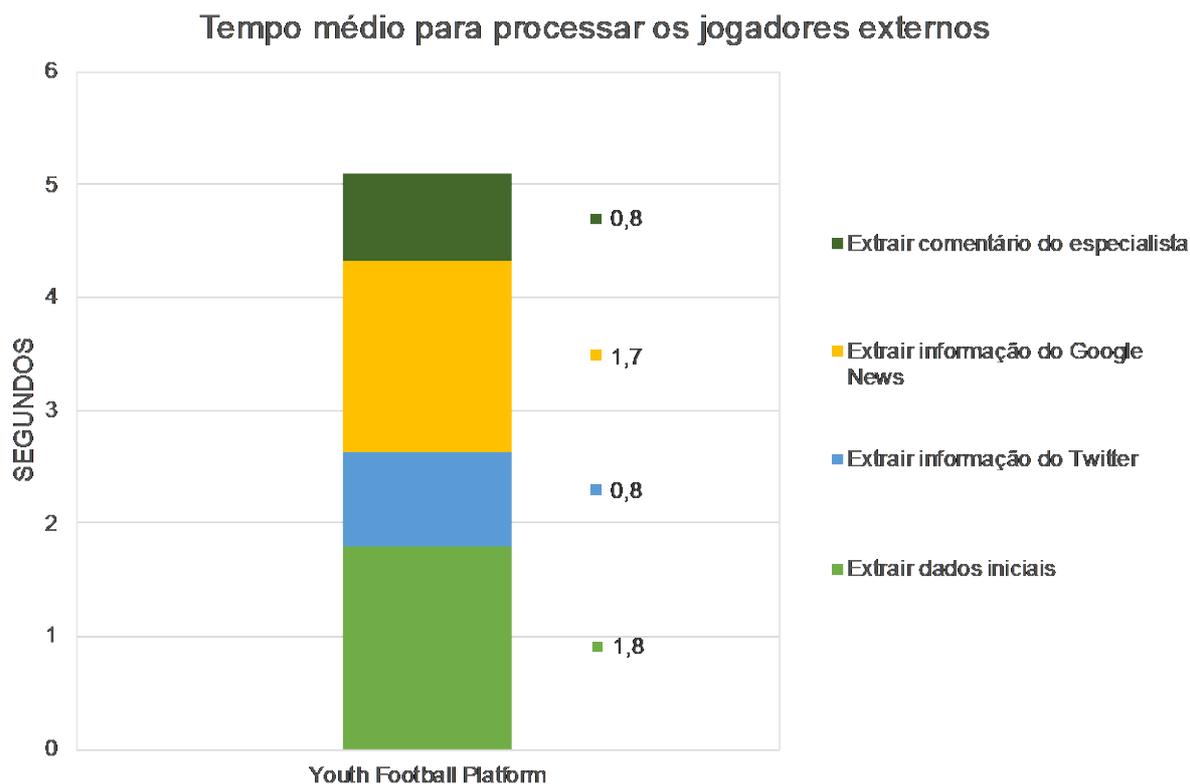


Figura 43. Tempo médio para processar os jogadores externos

Em suma, com os testes realizados à plataforma é possível concluir que o requisito funcional que demora mais tempo a processar é o requisito “Consultar jogadores externos”, com um tempo médio de 5,5 segundos. Isto deve-se ao facto de serem necessárias quatro extrações (três de fontes diferentes) para exibir a página. Por outro lado, os requisitos que demoram menos tempo a processar são, para além dos requisitos que não necessitam de nenhuma extração, os que extraem ou inserem informação na base de dados da plataforma, tais como “Consultar estatísticas internas”, “Consultar jogadores internos”, “Registrar na plataforma”, “Fazer login” e “Inserir jogadores na base de dados”, com um tempo de processamento imediato.

5. Conclusões e trabalho futuro

5.1. Conclusões

Num desporto cada vez mais competitivo e mais importante na sociedade, é inquestionável que os jogadores representam o ativo mais importante dos clubes. Assim, quem está à frente dos clubes deve reconhecer a importância de potenciar os seus jogadores, acrescentando valor tanto ao clube como ao jogador, beneficiando ambas as partes.

O futebol em Portugal está a passar a melhor fase de sempre em termos de exportação de jovens promissores, com o Sport Lisboa e Benfica, Futebol Clube do Porto e Sporting Clube de Portugal a contribuírem na sua maioria.

No entanto, para manter esta fase e elevarmos o nome de Portugal no futebol não basta olhar para os jovens mais velhos que já estão na elite do futebol, mas sim para todo o conjunto das camadas jovens, passando pelos benjamins até aos juniores nas diversas associações de futebol em Portugal.

Com o poder do dinheiro a imperar sobre qualquer organismo, os clubes sem possibilidades financeiras de investir num departamento de *scouting* com todas as ferramentas necessárias têm mais dificuldades de se potenciarem.

O estudo realizado ao longo do projeto permitiu concluir que há pouca informação sobre os clubes, jogadores e até mesmo resultados dos jogos relativamente ao futebol jovem em Portugal, sendo que a informação disponível está dispersa em diferentes plataformas. É importante contornar essa situação pois consoante a informação está mais dispersa, menos interesse é despertado pelos utilizadores, sendo os clubes portugueses de menor dimensão os principais prejudicados.

Os testes efetuados permitem concluir que a plataforma desenvolvida pode ser utilizada para ajudar a evoluir os clubes de menor dimensão e os jogadores menos conhecidos, assim como exibir informação diversa numa só plataforma.

O código desenvolvido para a criação da plataforma está disponível através do *link* <https://github.com/asrrs1-iscteulpt/YouthFootballPlatform>.

5.2. Trabalho futuro

A fase seguinte passa por melhorar o desempenho da recolha de dados de outras plataformas. Como demonstrado no capítulo 4, os tempos dos testes realizados ao desempenho da plataforma na extração de dados de alguns *sites* podem ser melhorados, pois quanto mais tempo uma página demora a carregar, menos interesse desperta no utilizador e a probabilidade de sair da

plataforma aumenta. Um bom exemplo de desempenho que é necessário seguir é o *web scraping* feito ao jornal *O Jogo* de forma a extrair as três últimas notícias sobre as camadas jovens, que leva cerca 0,3 segundos (Figura 40) a carregar a página. Comparativamente ao *web scraping* feito à *FPF* e ao *Zerozero* é bastante menos robusto, no entanto importa seguir o exemplo do seu desempenho.

A plataforma desenvolvida poderá ser aplicada a outras camadas jovens de modalidades que não tenham tanta visibilidade em Portugal, sendo necessária a análise de possíveis fontes de extração de dados dessas modalidades.

Algo que faltou implementar e poderá ser explorado são as plataformas de anotações desenvolvidas por académicos, pois poderá ser interessante agregar essa informação mais específica sobre cada jogador. No que toca às redes sociais, estas têm uma quantidade de informação imensurável e estão cada vez mais presentes na sociedade, podendo ser uma vertente a explorar ainda mais e a tirar o máximo do partido que estas trazem. Para além do *Twitter*, poderão ser exploradas outras redes sociais como o *Facebook*²³ e o *Instagram*²⁴.

Outro aspeto que poderá interessar é fazer um levantamento das maiores necessidades que os utilizadores sentem que estão em falta na plataforma, assim como as suas limitações. Uma das necessidades poderá ser a criação de diferentes permissões para os diferentes utilizadores na plataforma, pois a forma que o sistema foi desenvolvido dá as mesmas permissões a todos os utilizadores. De forma a solucionar essa necessidade, poderão ser criadas diferentes permissões e autorizações para cada categoria de utilizador: para os dirigentes poderá ser dada a permissão para inserir jogadores na base de dados, para os atletas poderá ser criada uma secção específica com mais dados sobre o jogador nas redes sociais, para os olheiros poderá ser criada uma secção que permita guardar uma lista de jogadores favoritos e para os adeptos poderá ser criado um painel de carregamento de vídeos e fotografias de jogos disputados.

Relativamente às limitações da plataforma, uma delas poderá ser o sistema de pesquisa de jogadores externos, pois nem sempre corresponde ao jogador pretendido e pode impactar diretamente na plataforma. Isto acontece devido ao resultado que o *Zerozero* apresenta em primeiro lugar quando é feita a pesquisa do jogador, pois o *web scraping* feito é sempre ao primeiro resultado que aparece. Esta limitação faz com que a plataforma por vezes demonstre um jogador que na realidade não foi o procurado. Um bom exemplo de um bom sistema de pesquisa de jogadores é o da própria plataforma (jogadores internos), pois para além da excelente performance, retorna o resultado que procuramos devido à forma como foi

²³ <https://www.facebook.com/>

²⁴ <https://www.instagram.com/>

desenvolvido, embora necessite de alterações ao código de forma a solucionar os *outputs* obtidos na secção 4.1.

Referências

- Aelieve. (2019). Best Soccer Websites. Retrieved from <https://aelieve.com/rankings/websites/category/sports/best-soccer-websites/>
- Ameya, N., Anil, P., & Dikshay, P. (2013). Type of NOSQL databases and its comparison with relational databases. *International Journal of Applied Information Systems*, 5(4), 16–19.
- Andrade, D. (2020). Benfica volta a perder na final da Youth League. *Público*. Retrieved from <https://www.publico.pt/2020/08/25/desporto/noticia/benfica-volta-perder-final-youth-league-1929248>
- Araújo de Morais, R. (2014). *Diários Desportivos em Portugal e Espanha: uma análise comparativa*. Universidade Fernando Pessoa.
- Bancada. (2019). Graças ao Football Manager... clube amador contrata experiente jogador francês. Retrieved from Bancada website: <https://bancada.pt/futebol/grandefutebol/gracas-ao-football-manager-clube-amador-contrata-experiente-jogador-frances>
- Barros, B. (2017). *Aplicação móvel distribuída para registo e avaliação em tempo real de desempenho de equipas desportivas* (ISCTE-IUL). Retrieved from <http://hdl.handle.net/10071/15166>
- Barros, B., Serrão, C., & Lopes, R. (2018). Distributed crowd-based annotation of soccer games using mobile devices. *IcSPORTS 2018 - Proceedings of the 6th International Congress on Sport Sciences Research and Technology Support*, 40–48. <https://doi.org/10.5220/0006927000400048>
- Batrinca, B., & Treleaven, P. C. (2014). Social media analytics: a survey of techniques, tools and platforms. *AI and Society*, 30(1), 89–116. <https://doi.org/10.1007/s00146-014-0549-4>
- Breuss, M. (2019). Beautiful Soup: Build a Web Scraper With Python. Retrieved from Real Python website: <https://realpython.com/beautiful-soup-web-scraper-python/>
- Browning, B., & Sanderson, J. (2016). The Positives and Negatives of Twitter: Exploring How Student-Athletes Use Twitter and Respond to Critical Tweets. *International Journal of Sport Communication*, 5(4), 503–521. <https://doi.org/10.1123/ijsc.5.4.503>
- Cadeirinhas, D. (2019). *diScout – Aplicação móvel distribuída para avaliação e recrutamento de jogadores de Futebol de formação*. ISCTE-IUL.
- CherryPy. (2020). CherryPy. Retrieved from <https://cherrypy.org/>
- Clemente, F. M. (2019). *Ultimate Performance Analysis Tool (uPATO) Implementation of Network Measures Based on Adjacency Matrices for Team Sports*.

- Correia, S. A., & da Silva, A. R. (2004). *Técnicas para Construção de Testes Funcionais Automáticos*. Retrieved from https://www.researchgate.net/profile/Alberto_Silva2/publication/252369706_Tecnicas_para_Construcao_de_Testes_Funcionais_Automaticos/links/0046352cd9ed7081c2000000.pdf
- Couvellaere, V., & Richelieu, A. (2005). Brand Strategy in Professional Sports: The Case of French Soccer Teams. *European Sport Management Quarterly*, 5(1), 23–46. <https://doi.org/10.1080/16184740500089524>
- DataFlair. (2020). Why MongoDB – 10 Reasons to Learn MongoDB for 2019. Retrieved from 2019 website: <https://data-flair.training/blogs/why-mongodb/>
- DB-Engines. (2020a). DB-Engines Ranking. Retrieved from <https://db-engines.com/en/ranking>
- DB-Engines. (2020b). *Relational DBMS*. Retrieved from <https://db-engines.com/en/article/Relational+DBMS?ref=RDBMS>
- Dijkshoorn, C. (2012). *Automatic Extraction of Soccer Game Events from Twitter*. Retrieved from https://www.researchgate.net/publication/267263735_Automatic_Extraction_of_Soccer_Game_Events_from_Twitter
- Ehmer, M., & Khan, F. (2012). A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. *International Journal of Advanced Computer Science and Applications*, 3(6). <https://doi.org/10.14569/ijacsa.2012.030603>
- Flask Documentation. (2010). Retrieved March 22, 2020, from <https://flask.palletsprojects.com/en/1.1.x/>
- Football-Index. (2020). Best Football Stats Sites. Retrieved from <https://football-index.com/stats/>
- Football Manager. (2005). Football Manager. Retrieved from <https://www.footballmanager.com>
- Guru99. (2020a). *Functional Testing Vs Non-Functional Testing: What's the Difference?* Retrieved from <https://www.guru99.com/functional-testing-vs-non-functional-testing.html>
- Guru99. (2020b). NoSQL Tutorial: Learn NoSQL Features, Types, What is, Advantages. Retrieved from <https://www.guru99.com/nosql-tutorial.html>
- Guru99. (2020c). *What is Selenium? Introduction to Selenium Automation Testing*. Retrieved from <https://www.guru99.com/introduction-to-selenium.html>

- Habib, O. (2016). Python Frameworks: Full-Stack vs. Micro Framework. Retrieved from <https://dzone.com/articles/python-frameworks-full-stack-vs-micro-framework>
- Hoffman, C. (2018). What Is an API? Retrieved from How-To Geek website: <https://www.howtogeek.com/343877/what-is-an-api/>
- Huggins, J. (2004). Selenium. Retrieved from <https://www.selenium.dev/>
- InStat. (2007). inStat. Retrieved from <https://instatsport.com>
- Introduction to Flask. (2008). Retrieved March 22, 2020, from <https://pymbook.readthedocs.io/en/latest/flask.html>
- Jayaram, P. (2016). When to Use (and Not to Use) MongoDB.
- JetBrains. (2019). The State of Developer Ecosystem Survey in 2018. Retrieved from <https://www.jetbrains.com/research/devecosystem-2018/python/>
- Lacerda, D. P., Dresch, A., Proença, A., & Antunes Júnior, J. A. V. (2013). Design Science Research: Método de pesquisa para a engenharia de produção. *Gestao e Producao*, 20(4), 741–761. <https://doi.org/10.1590/S0104-530X2013005000014>
- Lee, D. Y., Saha, R., Khan Yusufi, F. N., Park, W., & Karimi, I. A. (2009). Web-based applications for building, managing and analysing kinetic models of biological systems. *Briefings in Bioinformatics*, 10(1), 65–74. <https://doi.org/10.1093/bib/bbn039>
- MaisFutebol. (2016). Hoffenheim descobriu Roberto Firmino no Football Manager. Retrieved from MaisFutebol website: <https://maisfutebol.iol.pt/incrivel/alemanha/hoffenheim-descobriu-roberto-firmino-no-football-manager>
- Makai, M. (2020a). Databases. Retrieved from <https://www.fullstackpython.com/databases.html>
- Makai, M. (2020b). TurboGears. Retrieved from <https://www.fullstackpython.com/turbogears.html>
- Marcel Hellkamp. (2020). Bottle: Python Web Framework. Retrieved from <https://bottlepy.org/docs/dev/>
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266. [https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2)
- Mitchell, R. (2018). Web Scraping with Python, 2nd Edition. In *Web Scraping with Python, 2nd Edition*. <https://doi.org/10.1017/CBO9781107415324.004>
- MongoDB. (2020a). The most popular database for modern apps. Retrieved from <https://www.mongodb.com>
- MongoDB. (2020b). What Is A Non Relational Database. Retrieved from

- <https://www.mongodb.com/scale/what-is-a-non-relational-database>
- Oliveira, V. R. (2018). Portugal lidera futebol jovem na Europa dos últimos cinco anos. *TSF*. Retrieved from <https://www.tsf.pt/desporto/portugal-lidera-futebol-jovem-na-europa-ha-cinco-anos-9660074.html>
- Opta. (1996). Opta. Retrieved December 17, 2019, from <http://www.optasports.com>
- Oracle. (2020a). Database. Retrieved from <https://www.oracle.com/database/what-is-database.html>
- Oracle. (2020b). What Is a Relational Database? Retrieved from <https://www.oracle.com/database/what-is-a-relational-database/>
- Pallets Projects. (2020). Flask. Retrieved March 22, 2020, from <https://palletsprojects.com/p/flask/>
- Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D., & Giannotti, F. (2019). A public data set of spatio-temporal match events in soccer competitions. *Scientific Data*, 6(1), 236. <https://doi.org/10.1038/s41597-019-0247-7>
- Parker, Z., Poe, S., & Vrbsky, S. V. (2013). Comparing NoSQL MongoDB to an SQL DB. *Proceedings of the Annual Southeast Conference*. <https://doi.org/10.1145/2498328.2500047>
- Pedersen, P. M. (2014). A Commentary on Social Media Research From the Perspective of a Sport Communication Journal Editor. *Communication & Sport*, 2(2), 138–142. <https://doi.org/10.1177/2167479514527428>
- Peffer, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=epref&AN=JMIS.BD.DE.PEFFERS.DSRMIS&site=ehost-live> DP - EBSCOhost DB - epref
- Pegoraro, A., & Jinnah, N. (2012). Tweet 'em and reap “em: The impact of professional athletes” use of Twitter on current and potential sponsorship opportunities. *International Journal of Sport Communication, American Behavioral Scientist Journal of Sport Management, International Journal of Sport Marketing and Management, the Journal of Sponsorship, JOURNAL OF BRAND STRATEGY*, 1(1), 85–97.
- Petlovana, Y. (2020). Top 13 Python Web Frameworks to Learn in 2020. Retrieved from <https://steelkiwi.com/blog/top-10-python-web-frameworks-to-learn/>
- Politowski, C., & Maran, V. (2014). *Comparação de Performance entre PostgreSQL e MongoDB*.

- Price, J., Farrington, N., & Hall, L. (2013). Changing the game? The impact of Twitter on relationships between football clubs, supporters and the sports media. *Soccer and Society*, 14(4), 446–461. <https://doi.org/10.1080/14660970.2013.810431>
- Python. (2020). Web Frameworks for Python. Retrieved from Python Software Foundation website: <https://wiki.python.org/moin/WebFrameworks>
- Record. (2019). João Félix vence o prémio Golden Boy de 2019. *Record*. Retrieved from <https://www.record.pt/internacional/paises/espanha/detalhe/joao-felix-vence-o-premio-golden-boy-de-2019>
- Roseiro, B. (2020). Como uma plataforma ajudou Fred na adaptação ao Manchester United (com a ajuda de Gilberto Silva). *Observador*. Retrieved from <https://observador.pt/2020/12/02/como-uma-plataforma-ajudou-fred-na-adaptacao-ao-manchester-united-com-a-ajuda-de-gilberto-silva/>
- Ruby on Rails. (2020). Ruby on Rails. Retrieved from https://guides.rubyonrails.org/getting_started.html
- Sanderson, J. (2011). *It's a Whole New Ballgame: How Social Media is Changing Sports*.
- Schwarze, K. (2020). Why Non-Relational Databases Are on the Rise. Retrieved from <https://www.udemy.com/blog/why-non-relational-databases-are-on-the-rise/>
- Sharma, A. (2020). Difference between SQL and NoSQL. Retrieved from <https://www.geeksforgeeks.org/difference-between-sql-and-nosql/>
- SimilarWeb. (2020a). Top Websites Ranking in Portugal. Retrieved from <https://www.similarweb.com/top-websites/portugal/category/sports/>
- SimilarWeb. (2020b). Top Websites Ranking in the world. Retrieved from <https://www.similarweb.com/top-websites/category/sports/soccer/>
- STATSports. (1981). STATSports. Retrieved from www.statsports.com
- TechnoLush. (2019). Micro Vs Full-Stack Frameworks. Retrieved from <https://www.technolush.com/blog/micro-vs-full-stack-frameworks>
- The Punters Page. (2020). The 8 Best Football Websites for Stats & Analysis.
- Tonar, J. (2013). How to apply the principles of three-tier architecture in a web integration project. Retrieved from Web integration website: <http://www.web-integration.info/en/blog/how-to-apply-the-principles-of-three-tier-architecture-in-a-web-integration-project/>
- uPato. (2020). uPATO App. Retrieved from https://play.google.com/store/apps/details?id=pt.ubi.di.upato&hl=en_US
- Vendite, L. L., Moraes, A. C. de, & Vendite, C. C. (2015). Scout no futebol: uma análise

- estatística. *Conexões*, 1(2), 183–194. <https://doi.org/10.20396/conex.v1i2.8638024>
- Vergeer, M., & Mulder, L. (2019). Football Players' Popularity on Twitter Explained: Performance on the Pitch or Performance on Twitter? *International Journal of Sport Communication*, 12(3), 376–396. <https://doi.org/10.1123/ijsc.2018-0171>
- Visual Paradigm. (2020). UML Association vs Aggregation vs Composition. Retrieved from <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/>
- web2py. (2020). About web2py. Retrieved from <http://www.web2py.com/init/default/what>
- Williams, J., Chinn, S. J., & Suleiman, J. (2014). The value of Twitter for sports fans. *Journal of Direct, Data and Digital Marketing Practice*, 16(1), 36–50. <https://doi.org/10.1057/dddmp.2014.36>
- Witkemper, C., Lim, C., & Waldburger, A. (2012). Social media and sports marketing: Examining the motivations and constraints of Twitter users. *Sport Marketing Quarterly*, 21(3), 170.
- Wyscout. (2004). Wyscout. Retrieved from <https://wyscout.com/>
- Zerozero. (2019). Zerozero - Manual de Colaboração. Retrieved from https://www.zerozero.pt/zz_colaboracao.pdf
- Zublenko, E. (2020). Why Django is the Best Web Framework for Your Project. Retrieved from <https://steelkiwi.com/blog/why-django-best-web-framework-your-project/>

Anexos

A. Requisitos funcionais

Os requisitos funcionais descritos abaixo servem de auxílio aos testes funcionais efetuados na secção 4.1 - Testes funcionais.

Obter e apresentar highlights	
Nº do requisito	1
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e são exibidos os highlights das últimas notícias das camadas jovens de futebol em Portugal.

Tabela 10. Requisito 1 – Obter e apresentar highlights

Consultar estatísticas	
Nº do requisito	2
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Estatísticas” no menu e são exibidos dois botões para aceder às estatísticas externas ou internas.

Tabela 11. Requisito 2 - Consultar estatísticas

Consultar estatísticas externas	
Nº do requisito	3
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Estatísticas” no menu. Após isso, procede para as “Estatísticas externas” e insere na barra de procura o jogador que pretende exibir, pressionando no final o botão “Pesquisar”.

Tabela 12. Requisito 3 - Consultar estatísticas externas

Consultar estatísticas internas	
Nº do requisito	4
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Estatísticas” no menu. Após isso, procede para as “Estatísticas internas” e insere na barra de procura o jogador que pretende exibir, pressionando no final o botão “Pesquisar”.

Tabela 13. Requisito 4 - Consultar estatísticas internas

Consultar classificação	
Nº do requisito	5
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Média
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Classificação” no menu e são exibidos dois botões para aceder à classificação da época atual ou à classificação de épocas passadas.

Tabela 14. Requisito 5 - Consultar classificação

Consultar classificação época atual	
Nº do requisito	6
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Média
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Classificação” no menu. Após isso, pressiona o botão “Classificação época atual”. Escolhe uma opção de cada menu <i>dropdown</i> , um para a associação de futebol e o outro para a categoria das camadas jovens, e pressiona o botão “Seguinte”. De seguida escolhe a competição que pretende exibir e pressiona o botão “Seguinte”.

Tabela 15. Requisito 6 - Consultar classificação época atual

Consultar classificação épocas passadas	
Nº do requisito	7
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Média
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Classificação” no menu. Após isso, pressiona o botão “Classificação épocas passadas”. Escolhe uma opção de cada menu <i>dropdown</i> , um para a época desportiva, um para a associação de futebol e o outro para a categoria das camadas jovens, e pressiona o botão “Seguinte”. De seguida escolhe a competição que pretende exibir e pressiona o botão “Seguinte”.

Tabela 16. Requisito 7 - Consultar classificação épocas passadas

Consultar jornadas	
Nº do requisito	8
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Média
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jornadas” no menu. Escolhe uma opção de cada menu <i>dropdown</i> , um para a associação de futebol e o outro para a categoria das camadas jovens, e pressiona o botão “Seguinte”. De seguida, escolhe a competição e a jornada que pretende exibir e pressiona o botão “Seguinte”.

Tabela 17. Requisito 8 - Consultar jornadas

Consultar informação de jogadores	
Nº do requisito	9
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu e são exibidos dois botões para aceder aos jogadores externos ou internos.

Tabela 18. Requisito 9 - Consultar informação de jogadores

Consultar jogadores externos	
Nº do requisito	10
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu. Após isso, procede para “Jogadores externos” e insere na barra de procura o jogador que pretende exibir, pressionando no final o botão “Pesquisar”.

Tabela 19. Requisito 10 - Consultar jogadores externos

Consultar informação do Twitter	
Nº do requisito	11
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu. Após isso, procede para “Jogadores externos” e insere na barra de procura o jogador que pretende exibir, pressionando no final o botão “Pesquisar”. Ao abrir a página são exibidos <i>tweets</i> relacionados com o jogador e, caso haja, <i>tweets</i> de especialistas.

Tabela 20. Requisito 11 - Consultar informação do Twitter

Consultar informação do Google News	
Nº do requisito	12
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu. Após isso, procede para “Jogadores externos” e insere na barra de procura o jogador que pretende exibir, pressionando no final o botão “Pesquisar”. Ao abrir a página são exibidas as notícias relacionadas com o jogador.

Tabela 21. Requisito 12 - Consultar informação do Google News

Consultar jogadores internos	
Nº do requisito	13
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu. Após isso, procede para “Jogadores internos” e insere na barra de procura o jogador que pretende exibir, pressionando no final o botão “Pesquisar”.

Tabela 22. Requisito 13 - Consultar jogadores internos

Registar na plataforma	
Nº do requisito	14
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu. Após isso, procede para “Jogadores internos” e pressiona o botão “Inserir novo jogador interno”. Na página de login seleciona a opção “Registar”.

Tabela 23. Requisito 14 - Registar na plataforma

Fazer Login	
N° do requisito	15
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado e não registado
Descrição	O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu. Após isso, procede para “Jogadores internos” e pressiona o botão “Inserir novo jogador interno”. Na página de login insere o utilizador e a respetiva palavra-passe e pressiona o botão “Login”.

Tabela 24. Requisito 15 - Fazer login

Inserir jogadores na base de dados	
N° do requisito	16
Autor	André Rouiller
Dispositivo	Mobile
Prioridade	Alta
Utilizador	Registado
Descrição	Apenas para utilizadores registados na plataforma. O utilizador entra na plataforma e seleciona a opção “Jogadores” no menu. Procede para “Jogadores internos” e pressiona o botão “Inserir novo jogador interno”. Na página de login insere o utilizador e a respetiva palavra-passe e pressiona o botão “Login”. Após isso, insere os dados do jogador e pressiona o botão “Inserir jogador”.

Tabela 25. Requisito 16 - Inserir jogadores na base de dados

B. Vídeos dos testes realizados à plataforma

De seguida estão os *links* dos vídeos de testes realizados à plataforma.

Teste às funcionalidades da plataforma com base nos requisitos funcionais	
Nº do teste	1
Autor	André Rouiller
Tipo	Funcional
Dispositivo	Telemóvel
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/EYBa0C2od_ZKtx2C2ooyix4BH0lMJxDWlWXVP7jRN3U0bw?e=IoQRmk

Tabela 26. Teste às funcionalidades da plataforma com base nos requisitos funcionais

Teste de caixa-preta	
Nº do teste	2
Autor	André Rouiller
Tipo	Funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/ESor2YKaBRVetQ22NEy24KwBFR0nXP8FqymHjk9POiHjzg?e=LDxTcD

Tabela 27. Teste de caixa-preta

Teste automatizado às estatísticas externas	
Nº do teste	3
Autor	André Rouiller
Tipo	Funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/EVguhAeDfOZNho71XuVwJu4BBVxieKHEuxAL-3liCD2twg?e=YIJif2

Tabela 28. Teste automatizado às estatísticas externas

Teste automatizado à classificação da época atual	
Nº do teste	4
Autor	André Rouiller
Tipo	Funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/Ec_YVJrB9_JDrLFPPoYMraIBUn3ZxYyAi9W8c2N34eVGnw?e=9heu2m

Tabela 29. Teste automatizado à classificação da época atual

Teste automatizado à classificação de épocas passadas	
Nº do teste	5
Autor	André Rouiller
Tipo	Funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/EYzvM2mJafRHlnb2BrNiVu8BPLryeiHS1FKE8QzU_x2Ybw?e=inPhTs

Tabela 30. Teste automatizado à classificação de épocas passadas

Teste automatizado aos jogadores externos	
Nº do teste	6
Autor	André Rouiller
Tipo	Funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/ESgCy1M6rLZArSavGctQv2gBnFTkqPpD-u4i5ca3y6zWcw?e=irw9lj

Tabela 31. Teste automatizado aos jogadores externos

Teste de caixa-branca às estatísticas externas	
Nº do teste	7
Autor	André Rouiller
Tipo	Não funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/EXfAqWHvSSF0hmnr4KaqfN8Bp6Ekn5OAgz2TTDfjLdeU9A?e=b3QltQ

Tabela 32. Teste de caixa-branca às estatísticas externas

Teste de caixa-branca à classificação da época atual	
Nº do teste	8
Autor	André Rouiller
Tipo	Não funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/EWVMgFH0HjBFshUEc-CAvGkBSr6GUpN8cmfnCGwRG_rceA?e=NUhyJo

Tabela 33. Teste de caixa-branca à classificação da época atual

Teste de caixa-branca à classificação de épocas passadas	
Nº do teste	9
Autor	André Rouiller
Tipo	Não funcional
Dispositivo	Computador
Link	https://iscteiul365-my.sharepoint.com/:v:/g/personal/asrrs1_iscteiul_pt/EeYqwooXrHBJscVjWNRFrccBaisM9GzGkDwVx4oSbdWSg?e=h8a31a

Tabela 34. Teste de caixa-branca à classificação de épocas passadas

Teste de caixa-branca aos jogadores externos	
Nº do teste	10
Autor	André Rouiller
Tipo	Não funcional
Dispositivo	Computador
Link	https://iscteiu1365-my.sharepoint.com/:v:/g/personal/asrrs1_iscte-iul_pt/EQhDgKCGAEILiF09PndgoToBVV5VYBsZZPNxDt2NSIkRlg?e=a4sEGN

Tabela 35. Teste de caixa-branca aos jogadores externos

C. Código dos testes automatizados

Neste anexo está o código desenvolvido para realizar os testes automatizados.

```

TESTE_AUTOMATIZADO = {
1: 'Gonçalo+Batalha', 2: 'Gonçalo+Ramos', 3: 'Francisco+Conceição', 4: 'Rafael+Fernandes',
5: 'Rodrigo+Ferreira', 6: 'Nuno+Mendes', 7: 'Joelson+Fernandes', 8: 'Bernardo+Folha', 9: 'Tiago+Ferreira',
10: 'Leonardo+Buta', 11: 'Francisco+Lemos', 12: 'Miguel+Falé', 13: 'Carlos+Borges', 14: 'Jorge+Meireles',
15: 'Marco+Cruz', 16: 'Hugo+Félix', 17: 'Tiago+Augusto', 18: 'David+Costa', 19: 'David+Vinhas',
20: 'João+Tomé', 21: 'Fábio+Silva', 22: 'Ruben+Dias', 23: 'Tomás+Esteves', 24: 'Tiago+Tomás',
25: 'Yoann+Gonçalves', 26: 'Luís+Gomes', 27: 'Paulo+Bernardo', 28: 'Jordan+Gonçalves', 29: 'Ronaldo+Camará',
30: 'Tiago+Carvalho', 31: 'Rodrigo+Gomes', 32: 'Fernando+Schmelz', 33: 'Herculano+Nabian', 34: 'Chico+Lamba',
35: 'Rafael+Moreira', 36: 'Leonardo+Buta', 37: 'Henrique+Araújo', 38: 'Diogo+Cardoso', 39: 'Diogo+Quade',
40: 'Vasco+Gama', 41: 'Marco+Cruz', 42: 'Hugo+Neves', 43: 'Mauro+Ribeiro', 44: 'Kiko+Félix',
45: 'Gabriel+Carvalho', 46: 'Rodrigo+Melo', 47: 'Lara+Pintassilgo', 48: 'José+Bica', 49: 'Mateus+Fernandes',
50: 'Diogo+Carvalheira'
}

```

Figura 44. Ordem dos nomes dos jogadores procurados

```

from selenium import webdriver
from selenium.webdriver.common.by import By
import time

print('----- INÍCIO DO TESTE AUTOMATIZADO (ESTATÍSTICAS EXTERNAS) -----')
count = 1
path = "D:/NB25939/Documents/chromedriver"
driver = webdriver.Chrome(path)
while count <= 50:
    try:
        antes = time.time()
        driver.get("http://192.168.1.228:8000/statistics?search=%s" %(TESTE_AUTOMATIZADO[count]))
        depois = time.time()
        tempo = round(depois-antes, 3)
        nome_jogador = driver.find_element(By.XPATH, '/html/body/div[4]/table/tbody/tr[1]/td/div/b')
        print('Teste nº %s: %s | Duração: %s' %(count, nome_jogador.text, tempo))
    except:
        print('Teste nº %s: Erro!' %(count))
        count += 1
        time.sleep(5)
driver.quit()
print('----- FIM DO TESTE AUTOMATIZADO (ESTATÍSTICAS EXTERNAS) -----')

```

Figura 45. Código-fonte do teste automatizado para as estatísticas externas

```

from selenium import webdriver
from selenium.webdriver.common.by import By
import time

print('----- INÍCIO DO TESTE AUTOMATIZADO (JOGADORES EXTERNOS) -----')
count = 1
path = "D:/NB25939/Documents/chromedriver"
driver = webdriver.Chrome(path)
while count <= 50:
    try:
        antes = time.time()
        driver.get("http://192.168.1.228:8000/players?search=%s" %(TESTE_AUTOMATIZADO[count]))
        depois = time.time()
        tempo = round(depois-antes, 3)
        nome_jogador = driver.find_element(By.XPATH, '/html/body/div[4]/table[1]/tbody/tr/td/b')
        print('Teste nº %s: %s | Duração: %s' %(count, nome_jogador.text, tempo))
    except:
        print('Teste nº %s: Erro!' %(count))
        count += 1
        time.sleep(5)
driver.quit()
print('----- FIM DO TESTE AUTOMATIZADO (JOGADORES EXTERNOS) -----')

```

Figura 46. Código-fonte do teste automatizado para os jogadores externos

```

from selenium import webdriver
from selenium.webdriver.common.by import By
import time

print('----- INÍCIO DO TESTE AUTOMATIZADO (CLASSIFICAÇÃO ÉPOCA ATUAL) -----')
count = 1
path = "D:/NB25939/Documents/chromedriver"
driver = webdriver.Chrome(path)
while count <= 50:
    try:
        antes_scraping1 = time.time()
        driver.get("http://192.168.1.228:8000/classification?af=af_braganca&campeonato=juvenis")
        depois_scraping1 = time.time()
        tempo_scraping1 = round(depois_scraping1-antes_scraping1, 3)

        seguinte = driver.find_element(By.XPATH, '/html/body/div[4]/form/input[1]')

        antes_scraping2 = time.time()
        seguinte.click()
        depois_scraping2 = time.time()
        tempo_scraping2 = round(depois_scraping2-antes_scraping2, 3)
        tempo_total = round(tempo_scraping1 + tempo_scraping2, 3)

        print('Teste nº %s: %s // Duração scraping 1: %s // Duração scraping 2: %s' %(count,
            'Campeonato Distrital Juvenis | Petrocavaleiros', tempo_scraping1, tempo_scraping2))
        print('Teste nº %s: Duração total: %s' %(count, tempo_total))
    except:
        print('Teste nº %s: Erro!' %(count))
        count += 1
        time.sleep(5)
driver.quit()
print('----- FIM DO TESTE AUTOMATIZADO (CLASSIFICAÇÃO ÉPOCA ATUAL) -----')

```

Figura 47. Código-fonte do teste automatizado para a classificação da época atual

```

from selenium import webdriver
from selenium.webdriver.common.by import By
import time

print('----- INÍCIO DO TESTE AUTOMATIZADO (CLASSIFICAÇÃO ÉPOCAS PASSADAS) -----')
count = 1
path = "D:/NB25939/Documents/chromedriver"
driver = webdriver.Chrome(path)
while count <= 50:
    try:
        antes_scraping1 = time.time()
        driver.get("http://192.168.1.228:8000/archive?epoca=16_17&af=af_braganca&campeonato=juvenis")
        depois_scraping1 = time.time()
        tempo_scraping1 = round(depois_scraping1-antes_scraping1, 3)

        seguinte = driver.find_element(By.XPATH, '/html/body/div[4]/form/input[1]')

        antes_scraping2 = time.time()
        seguinte.click()
        depois_scraping2 = time.time()
        tempo_scraping2 = round(depois_scraping2-antes_scraping2, 3)
        tempo_total = round(tempo_scraping1 + tempo_scraping2, 3)

        print('Teste nº %s: %s // Duração scraping 1: %s // Duração scraping 2: %s' %(count,
            'C.D. JUVENIS', tempo_scraping1, tempo_scraping2))
        print('Teste nº %s: Duração total: %s' %(count , tempo_total))
    except:
        print('Teste nº %s: Erro!' %(count))
        count += 1
        time.sleep(5)
driver.quit()
print('----- FIM DO TESTE AUTOMATIZADO (CLASSIFICAÇÃO ÉPOCAS PASSADAS) -----')

```

Figura 48. Código-fonte do teste automatizado para a classificação de épocas passadas