INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Análise do Movimento dos Atletas em Eventos Futebolísticos

JOÃO MANUEL COSTA CAMPOS
Julho de 2023

P.PORTO

Análise do Movimento dos Atletas em Eventos Futebolísticos

João Manuel Costa Campos

Aluno nº: 1180597

Dissertação para obtenção do Grau de

Mestre em Engenharia de Inteligência Artificial

Orientador: Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto
Supervisor Externo: Luís Telmo Soares Costa, AI & ML Engineer na DevScope

Júri:
Presidente:

Doutor Luiz Felipe Rocha de Faria, Professor Coordenador do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Vogais:

Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Doutor Paulo Sérgio dos Santos Matos, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Porto, julho de 2023

"O insucesso é apenas uma oportunidade para recomeçar com mais inteligência"

Henry Ford

Resumo

O mercado do futebol está em alta, com jogadores e treinadores sendo cada vez mais valorizados. Para garantir um desempenho superior, é crucial fazer escolhas criteriosas na contratação. Além disso, há uma demanda crescente por dados nesse setor, e métricas avançadas, como "expected goals", estão a tornar-se populares na análise de jogos de futebol. Essas métricas, originalmente usadas por mercados de apostas, agora são adotadas por comentadores e treinadores renomados. Isso indica que a análise de dados é essencial para melhorar o desempenho de todos os envolvidos no futebol.

Diante desse cenário, surge a necessidade de desenvolver uma solução que consiga explorar sequências e padrões de jogo através de análises avançadas e consiga extrair padrões de jogo a partir de imagens de sequências.

A metodologia utilizada neste projeto de pesquisa é a *Design Science Research.* Inicialmente, foi realizada uma revisão bibliográfica sobre os tipos de dados existentes no contexto do futebol, as métricas avançadas atualmente em alta no mundo analítico desportivo e soluções existentes no ramo. Foram identificadas e descritas algumas das características e limitações mais comuns dos serviços atuais do mercado. Este trabalho pretende apresentar uma proposta que inove no cálculo da métrica de xG, consiga identificar diversas estatísticas calculadas a partir de dados de eventos e consiga estabelecer uma relação entre esses dados, as sequências das equipas e o estilo de jogo da equipa.

O sistema Verance App utiliza dados do tipo de fluxo de eventos para calcular estatísticas para todas as equipas que atuaram nas principais 6 ligas durante a presente temporada (2022/23) e apresentar estatísticas de todas as sequências e ações destas mesmas equipas. Para além disto, apresenta também a funcionalidade de apresentação das 3 equipas mais semelhantes em análise.

A Verance App não foi utilizada por nenhuma equipa real para fornecer informação de melhoria dos resultados desportivos, mas foi avaliada tendo em conta os seus 3 componentes principais, o modelo xG, o modelo xT e a componente de extração dos padrões das sequências. A análise confirma que a solução projetada, na maioria das circunstâncias, apresenta resultados superiores aos dos serviços atuais do mercado.

Palavras-chave (Tema): ***xG*, *xT*, *Futebol, Estatísticas Avançadas, Inteligência Artificial, Aprendizagem Profunda***

Palavras-chave (Tecnologias): **Python, Pytorch, Mediator, React**

Abstract

The football market is booming, with players and coaches being increasingly valued. To ensure superior performance, it is crucial to make careful choices in recruitment. Additionally, there is a growing demand for data in this industry, and advanced metrics like "expected goals" are becoming popular in football analysis. These metrics, originally used by betting markets, are now adopted by renowned commentators and coaches. This indicates that data analysis is essential for improving the performance of everyone involved in football.

In this context, there is a need to develop a solution that can explore game sequences and patterns through advanced analysis and extract patterns from sequence images.

The methodology used in this research project is Design Science Research. Initially, a literature review was conducted on the types of data available in the context of football, the currently popular advanced metrics in sports analytics, and existing solutions in the field. Some of the common characteristics and limitations of current market services were identified and described. This work aims to propose an innovative approach to calculating the xG metric, to identify various statistics derived from event data, and to establish a relationship between this data, team sequences, and team playing style.

The Verance App system utilizes event stream data to calculate statistics for all teams participating in the top 6 leagues during the current season (2022/23) and presents statistics for all sequences and actions of these teams. Additionally, it provides the functionality to present the top 3 most similar teams for analysis.

The Verance App has not been used by any real team to provide performance improvement insights. However, it has been evaluated based on its three main components: the xG model, the xT model, and the sequence pattern extraction component. The analysis confirms that, in most circumstances, the designed solution outperforms the current market services.

Keywords (Theme): ***xG*, *xT*, *Soccer, Advanced Stats, Artificial Intelligence, Deep Learning***

Keywords (Technology): **Python, Pytorch, Mediator, React**

Agradecimentos

Gostaria de expressar a minha sincera gratidão a todos os que estiveram ao meu lado ao longo desta jornada. Agradeço de coração à minha família, cujo amor e apoio incondicionais foram fundamentais para alcançar este marco na minha vida. Aos meus amigos e colegas de faculdade, agradeço pela companhia, incentivo e momentos inesquecíveis compartilhados ao longo destes anos.

Gostava, também, de expressar o meu profundo agradecimento a algumas pessoas em particular. Agradeço imensamente ao Luís Costa pela sua orientação e supervisão durante este projeto. A sua expertise e apoio foram fundamentais para o meu crescimento profissional e pessoal. Também quero agradecer aos meus colegas mais próximos da empresa que me ajudaram como puderam com os seus conselhos, sendo eles André Reis e Rúben Teixeira.

Por fim, obrigado ao professor Dr. Constantino Martins, meu orientador, pelo seu tempo, correções e valiosas contribuições ao projeto. A sua experiência e orientação foram de extrema importância para o sucesso do mesmo.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Listagens

Notação e Glossário

API	Interface de Aplicação (do inglês *Application Programming Interface*)
DL	Aprendizagem Profunda (do inglês *Deep Learning*)
DSR	Metodologia de Investigação (*Design Science Research*)
ER	Engenharia de Requisitos
ETL	Extração, Transformação e Carregamento (do inglês *Extract, Transform, Load*)
FID	Distância inicial de Fréchet (do inglês *Frechet Inception Distance*)
FURPS	Funcionalidade Usabilidade Reliabilidade Performance Suportabilidade
GAN	Rede Adversária Generativa (do inglês *Generative Adversarial Network*)
GPU	Unidade de Processamento Gráfico (do inglês *Graphics Processing Unit*)
HTML	Linguagem de Programação (*HyperText Markup Language*)
IA	Inteligência Artificial
ID	Identificador
IEEE	Base de Dados de Relatórios Científicos (*Institute of Electrical and Electronics Engineers*)
ISEP	Instituto Superior de Engenharia do Porto
JSON	Formato de Ficheiro. (*JavaScript Object Notation*)
ML	Aprendizagem Automática (do inglês *Machine Learning*)
MSE	Erro Quadrático Médio (do inglês *Mean Squared Error*)
PRISMA	Metodologia de Pesquisa (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*)
RAE-L2	Arquitetura de Autoencoder. (*Regularized Autoencoder with L2-norm*)
RF	Requisito Funcional
RGPD	Regulamento Geral sobre a Proteção de Dados
RNA	Rede Neuronal Artificial

t-SNE	Técnica de Redução de Dimensionalidade. (*t-Distributed Stochastic Neighbor Embedding*)
UC	Caso de Uso (do inglês Use Case)
VAE	Autoencoder Variacional (do inglês *Variational Autoencoder*)
VAEGAN	Arquitetura de Modelo. (*Variational Autoencoder Generative Adversarial Network*)
VAEP	Métrica avançada. (*Valuing Actions by Estimating Probabilities*)
xA	Métrica de Assistência Esperada (do inglês *Expected Assist*)
xG	Métrica de Golo Esperado (do inglês *Expected Goal*)
xOVA	Métrica de Valor Ofensivo Esperado Adicionado (do inglês *Expected Offensive Value Added*)
xT	Métrica de Perigo Esperado (do inglês *Expected Threat*)

1 Introdução

Neste capítulo são identificados o problema que se pretende solucionar com este projeto, os objetivos a serem atingidos, as motivações para o seu desenvolvimento e os contributos aquando da sua finalização. Por fim, é apresentada a metodologia de investigação utilizada no desenvolvimento do projeto.

1.1 Contexto do Problema

O mercado do futebol nunca esteve tão valorizado como nos dias que correm (Paraeles, 2019; Sportinforma, 2019). Jogadores e treinadores são vendidos cada vez mais caros e, como tal, deverá haver melhor critério na sua escolha, de forma a obter uma melhor performance dos mesmos.

Por outro lado, há uma demanda maior para obtenção de dados deste ramo (SBE, 2021; SciSports, 2021). Métricas avançadas, como "*expected goals*", estão a tornar-se mais populares para analisar jogos de futebol (Whitmore, 2021c). Estas métricas foram usadas primeiramente por mercados de apostas, mas atualmente são usadas por comentadores e treinadores de futebol renomados (Whitmore, 2021c). Com isto, é possível deduzir que o mundo da análise de dados é o caminho a seguir para aumentar a probabilidade de uma melhor performance por parte de todos os intervenientes.

Surge assim a necessidade de desenvolver uma solução que através de técnicas de *Machine Learning* (ML) consiga explorar a potencialidade de sequências e padrões de jogo através de métricas avançadas (*expected threat* e *expected goals*), e através de métodos não

supervisionados de *Deep Learning* (DL), permita extrair padrões de jogo através de imagens de sequências.

1.2 Objetivos

O objetivo principal é o uso e implementação de algoritmos e técnicas de DL para melhorar a análise de futuros adversários ou auxiliar a contratação de um novo treinador e jogadores.

A questão de investigação é, portanto, a seguinte:

Será possível, através de técnicas de DL, a partir de dados de eventos de futebol conseguir extrair sequências e padrões de jogo de equipas de futebol?

Como tarefas para concretizar o objetivo, têm-se:

- Estudar e sintetizar os conhecimentos acerca de ML e DL na análise de dados, mais concretamente na área do futebol;
- Construir um algoritmo que, a partir de dados de eventos futebolísticos, consiga retornar sequências;
- Construir um modelo que classifica as sequências, de acordo com os seus padrões, detetadas no ponto anterior;
- Construir um modelo de cálculo de *expected goals;*
- Construir um modelo de cálculo de *expected threats;*
- Testar e avaliar o desempenho dos modelos e da solução implementada.

1.3 Motivação

Uma das principais motivações é o desejo de melhorar o desempenho de atletas e equipas. Ao analisar dados relacionados ao desempenho de um atleta ou equipa, treinadores e outros profissionais podem identificar pontos fortes e fracos e desenvolver estratégias direcionadas para a melhoria. Isso pode ajudar equipas e atletas a obter melhores resultados em campo e, finalmente, atingir os seus objetivos.

Outra motivação para desenvolver projetos de análise desportiva é o desejo de obter vantagem competitiva. No cenário desportivo atual, equipas e atletas procuram constantemente maneiras de ganhar vantagem sobre os seus adversários. Ao analisar os dados dos oponentes para entender melhor os fatores que afetam o desempenho, as equipas e os atletas podem tomar decisões informadas sobre como se preparar melhor e responder a diferentes cenários.

Além disto, há um crescente reconhecimento do valor dos *insights* orientados por dados na gestão e negócios desportivos (Analyisport, 2021). Proprietários, patrocinadores e outras partes interessadas, no ramo do futebol, estão cada vez mais à procura de formas de otimizar recursos e tomar decisões informadas como, por exemplo, a contratação de um jogador ou treinador. Ao usar a análise de dados para entender tendências e padrões de performance, equipas e organizações podem tomar decisões mais estratégicas e financeiramente mais sólidas.

No geral, a motivação para desenvolver projetos deste tipo vem do desejo de melhorar o desempenho, obter vantagem competitiva e tomar decisões informadas na indústria desportiva.

As motivações do autor para o desenvolvimento deste projeto vêm da sua enorme paixão pelo futebol e do desejo de perceber como as performances da sua equipa poderiam ser melhoradas.

1.4 Resultados Esperados

Atendendo à concretização do objetivo final, os resultados esperados deste projeto são os seguintes:

- Aplicação do sistema para melhorar a tomada de decisão na análise de futuros adversários. Sabendo quais os tipos de sequências mais utilizado pelo adversário, pode ser traçada uma estratégia para ajudar no confronto com um futuro adversário;

- Aplicação do sistema para auxílio na contratação de um novo treinador. Associando o tipo de sequências mais comum de um treinador a um estilo de jogo pretendido por uma equipa, pode ser usado para auxílio na contratação do mesmo;

- Análise das ferramentas existentes para análise de padrões de jogo em eventos futebolísticos. No estado da arte deste documento estão sintetizadas as soluções

existentes atualmente. São, também, expostas as suas funcionalidades, semelhanças e diferenças;

- Revisão da literatura nos campos das *advanced analytics* (*expected threat* e *expected goals*). No capítulo de Estado de Arte, são explicados conceitos relativos a métricas avançadas, apresentados os cálculos e exemplos;

- Análise exploratória de dados de eventos futebolísticos de um conjunto de partidas de futebol. Na fase de implementação, é feita uma análise exploratória dos dados e os seus resultados são expostos neste documento;

- Resultados do pré-processamento e processamento de técnicas levantadas a partir da revisão da literatura. Todo o pré-processamento e o uso de técnicas de ML e DL é exposto no capítulo 4 deste documento.

1.5 Metodologia de Investigação

A metodologia *Design Science Research* (DSR) é adequada para resolver problemas criando inovação (Cheong et al., 2013). Utiliza conhecimento adquirido para resolver problemas, criar mudança ou melhoria de soluções já existentes e gera novos conhecimentos, perceções e explicações teóricas (Baskerville et al., 2015; Horváth, 2007). Para além disto:

- Permite uma melhoria contínua, visto que o desenho da solução é incremental, permitindo, em qualquer fase do período experimental, fazer alterações e melhorias ao mesmo (Molina et al., 2007);

- Promove uma abordagem mais prática para a resolução dos problemas identificados, através de estratégias inovadoras que permitem que os utilizadores desenvolvam melhor as suas competências (Akker, 2006);

E todas as evidências que existem na literatura (Hevner et al., 2004; McKenney & Reeves, 2012), tornam esta metodologia a mais adequada e que melhor se encaixa neste projeto e, deste modo, será a utilizada.

A metodologia DSR geralmente inclui seis etapas ou atividades (Grenha Teixeira et al., 2016; Lapão et al., 2017; Peffers et al., 2007):

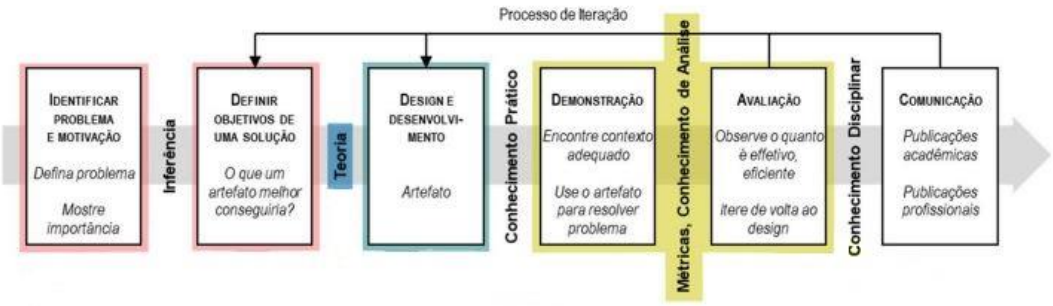


Figura 1: As 6 Atividades da Metodologia DSR (Pimentel & Filippo, 2020)

1. Identificação do problema, definição do problema de pesquisa e justificação do valor de uma possível solução;

2. Definição dos objetivos de uma solução;

3. Design e desenvolvimento de Artefactos (construções, modelos, algoritmos, métodos, etc.);

4. Demonstração fazendo uso do artefacto para resolver o problema em questão;

5. Avaliação da solução comparando os objetivos e os resultados reais com aqueles obtidos através do uso do artefacto;

6. Comunicação do problema, do artefacto, a sua utilidade e eficácia a outros pesquisadores e profissionais praticantes.

De notar que as pesquisas nem sempre precisam de começar pelo primeiro passo, fazendo desta uma metodologia flexível, mas principalmente que passe por todas as etapas identificadas (Peffers et al., 2007).

1.6 Estrutura do Documento

Este documento está dividido em 5 capítulos: Introdução, Estado da Arte, Análise e Desenho da Solução, Implementação e Avaliação da Solução e Conclusões.

O capítulo atual corresponde ao capítulo introdutório, cuja função é introduzir ao leitor o contexto do problema, os objetivos, a motivação, as contribuições para a área do futebol e da inteligência artificial e a metodologia de investigação utilizada no projeto.

O segundo capítulo apresenta a metodologia de pesquisa, introduz conceitos, técnicas e algoritmos utilizados na área de investigação, bem como expõe soluções existentes e trabalhos relacionados com o domínio em questão. A análise e desenho da solução, acompanhados de modelos e diagramas úteis, podem ser encontrados no terceiro capítulo.

O quarto capítulo descreve a implementação da solução proposta para este problema, estão presentes neste capítulo, ainda, todos os testes executados e validações efetuadas, de modo a garantir o funcionamento esperado da solução, e por fim é feita uma avaliação do resultado final de acordo com análises estatísticas e empíricas.

O relatório conta ainda com o quinto capítulo onde é feita uma descrição das tarefas concluídas, uma projeção de possíveis melhorias para trabalhos futuros e uma apreciação final sobre o trabalho desenvolvido.

2 Estado da Arte e Formalização Teórica

Este capítulo destina-se a apresentar um enquadramento teórico sobre os temas relacionados com este projeto, resumir alguns trabalhos nesta área e a expor as tecnologias existentes em torno do problema e da solução proposta. O conteúdo aqui presente é o resultado de diversas pesquisas e análises por parte do autor, de acordo com a metodologia de pesquisa também apresentada neste capítulo.

2.1 Metodologia de Pesquisa

PRISMA, "*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*", é uma metodologia transparente, confiável e amplamente utilizada para conduzir revisões sistemáticas e meta-análises (Page et al., 2021). É projetada para identificar, selecionar e sintetizar evidências de pesquisa relevantes para uma questão de investigação específica (Page et al., 2021; Rethlefsen et al., 2021).

A metodologia Prisma inclui as seguintes etapas (Page et al., 2021):

1. Definição da questão de Investigação: definição clara da questão de investigação e especificação dos critérios de inclusão e exclusão para os estudos a serem incluídos na revisão;

2. Pesquisa dos Artigos: etapa da pesquisa abrangente de estudos que abordam a questão de pesquisa usando uma variedade de bancos de dados e outras fontes;

3. Triagem dos Artigos: revisão dos títulos e resumos dos artigos identificados na pesquisa para determinar quais atendem aos critérios de inclusão e devem estar inclusos na revisão;

4. Seleção dos Artigos: leitura completa dos artigos e seleção dos que devem estar incluídos na revisão;

5. Extração de Informação dos Artigos: extração de dados relevantes dos estudos selecionados, incluindo informação sobre o desenho da solução, a amostra em estudo, o desenvolvimento e discussão dos resultados obtidos;

6. Avaliação da qualidade do Artigo: avaliação da qualidade dos estudos selecionados usando um conjunto predefinido de critérios, como a pertinência do estudo, os resultados obtidos ou a metodologia envolvida;

7. Sintetização dos resultados: resumo dos resultados selecionados, tendo em conta a qualidade e pertinência de cada um;

8. Interpretação dos resultados: interpretação dos resultados da revisão no contexto da questão de investigação e do corpo de conhecimento existente;

9. Escrita dos resultados: escrita dos resultados da revisão de forma clara e transparente, incluindo uma descrição detalhada dos métodos utilizados, os resultados obtidos e quaisquer limitações da revisão.

Neste projeto a metodologia de pesquisa utilizada foi uma abordagem simplificada de PRISMA. Com o objetivo de responder à questão de investigação identificada no subcapítulo “Objetivos” do capítulo anterior, foram definidas as seguintes questões:

Tabela 1: Questões Metodologia PRISMA

Identificador	Descrição
Q-01	Quais são os tipos de dados utilizados no Futebol?
Q-02	Quais são as métricas avançadas utilizadas no Futebol?
Q-03	Como são calculadas as métricas avançadas utilizadas no Futebol?
Q-04	O que é uma Sequência no contexto do Futebol?
Q-05	Como é calculada uma Sequência no contexto do Futebol?

Q-06	Que modelos podem ser utilizados para a extração das características das sequências?
Q-07	Quais são as soluções ou serviços existentes atualmente?

Apenas foram incluídos artigos escritos em inglês, de até 4 anos atrás, ou seja, o ano de 2019, que fossem relevantes para a questão de investigação definida.

A pesquisa foi efetuada principalmente através das bases de dados Springer Science, ResearchGate*, Institute of Electrical and Electronics Engineers* (IEEE) e ScienceDirect e foi conduzida por *keywords.* Depois de lido o título e resumo do artigo, este era selecionado se houvesse pertinência para tal. Depois da filtragem de artigos que tinham temas comuns, os que sobraram foram lidos na íntegra e usados na escrita do presente capítulo. Foi encontrado um total de 50 artigos. Após o fluxo descrito em cima, foram excluídos 31 artigos, ficando um total de 19 artigos.

2.2 Tipos de Dados no Contexto do Futebol

Para entender as escolhas feitas no desenvolvimento do projeto, é necessário entender primeiro os tipos de dados com os quais se trabalha no futebol, as empresas que os providenciam e em que casos de uso podem-se aplicar os mesmos.

No futebol, dados de eventos, "*event stream data*", são informações recolhidas e gravadas durante um jogo. Isso pode incluir informações como o resultado do jogo, o tempo restante de jogo, a localização da bola no campo, o tipo de evento e se esse lance foi realizado com sucesso ou não (Chandradas, 2021).

Como tipos de eventos temos, por exemplo, remate, drible, interceção, defesa, etc. O detalhe dos tipos de eventos depende da organização que providencia esses dados. Este tipo de dados é usado para ajudar a analisar o jogo e fornecer informações sobre o desempenho dos jogadores e equipas (Chandradas, 2021; Decroos et al., 2019; Merhej et al., 2021).

Os dados de rastreamento ótico, "*optical tracking data*", por outro lado, são informações registadas sobre os movimentos dos jogadores em campo (Track160, 2021). Está incluso, nestes dados, coisas como a distância percorrida por um atleta, a sua velocidade e a direção e intensidade dos seus movimentos. Os dados de rastreamento, geralmente, são usados em combinação com dados de evento para fornecer uma imagem mais detalhada do que está a

acontecer durante uma partida de futebol (Track160, 2021; Vidal-Codina et al., 2022). Após isto, a Q-01 foi respondida.

Existem muitas organizações diferentes que providenciam dados de eventos e dados de rastreamento no futebol. Algumas destas organizações mais conhecidas incluem a Opta e Prozone, que atualmente pertencem ao grupo (Stats Perform, 2022b) e (StatsBomb, 2022a). Estas empresas coletam e analisam dados de jogos de futebol e fornecem-nos a equipas, ligas, emissoras e outras organizações interessadas nessas informações. Outros fornecedores destes dados incluem empresas como (Wyscout, 2022), (InStat, 2022) e (ChyronHego, 2022).

Derivado desta diversidade de organizações surgem algumas dificuldades e desafios para o processamento destes dados, por parte de analistas (Decroos et al., 2019). Podem ser identificados 4 grandes desafios (Decroos et al., 2019):

1. Um desafio prende-se pelo facto de os dados servirem para múltiplos propósitos e objetivos, por exemplo uso por comentadores, jornais ou clubes de futebol, o que significa que a informação não está necessariamente desenhada para facilitar a sua análise. Nos dados de algumas organizações falta informação, tem informação irrelevante ou exige mais complexidade nos passos de pré-processamento. A Wyscout não regista a localização final de remates e separa os eventos entre duelos de dois jogadores, por exemplo;
2. O segundo desafio está relacionado com a terminologia e definição de alguns conceitos ser diferente dependendo da empresa que providencia os dados, o que significa que um *software* não pode ser usado para analisar a informação de dois fornecedores diferentes, sem os devidos ajustes de mapeamento.
3. Outro desafio está no facto de alguns vendedores destas informações já providenciarem dados há mais de uma década e serem incapazes de alterar as escolhas do design inicial desta informação. Houve evoluções na informação que alguns fornecedores anotam e agora incluem eventos adicionais ou informações mais detalhadas.
4. A última dificuldade que é possível identificar prende-se pelo extremo detalhe de informação opcional fornecida. A Opta, por exemplo, tem 4 tipos diferentes de remates, dependendo do seu resultado. Embora úteis, estas informações dificultam alguns passos de processamento e tornam extremamente difícil a aplicação de ferramentas de análise automática.

Ambos os tipos de dados são ferramentas importantes para treinadores e jogadores, que usam essas informações para melhorar o desempenho e obter vantagem competitiva e também para analistas e comentadores para tecer criticas nos seus comentários futebolísticos (Decroos et al., 2019).

2.3 Métricas Avançadas

As métricas avançadas, "*advanced analytics*", no futebol referem-se ao uso de técnicas complexas de análise de dados para extrair informações valiosas e melhorar o desempenho (Lichtenthaler, 2022). Essas técnicas podem incluir ML, modelos de previsão e visualização de dados e, geralmente, são obtidos através do uso de dados de eventos e dados de rastreamento para fornecer uma compreensão mais detalhada do jogo (Lichtenthaler, 2022; Pantzalis & Tjortjis, 2020).

Um exemplo de análise avançada no futebol é o uso de modelos de previsão para prever a probabilidade de uma equipa vencer um jogo com base em diversos fatores, como a localização da bola, o placar e os movimentos dos jogadores (Pantzalis & Tjortjis, 2020). Outro exemplo é o uso da visualização de dados para criar gráficos interativos que mostram os movimentos dos jogadores em campo, o que pode ajudar os treinadores e jogadores a identificar áreas onde podem melhorar o seu desempenho (Lichtenthaler, 2022; Pantzalis & Tjortjis, 2020).

De seguida, são explicadas algumas destas métricas avançadas, é exposto o seu cálculo e são apresentados exemplos.

2.3.1 xG

Os golos esperados, "*expected goals"* (xG), são uma medida estatística usada para avaliar a qualidade de uma oportunidade de golo no futebol. O número resultante indica quantos golos, em média, uma equipa esperaria marcar numa oportunidade semelhante. No cálculo, os analistas geralmente recorrem a técnicas estatísticas avançadas, como análise de regressão, de forma a conseguirem perceber a relação entre uma variedade de fatores e a probabilidade de um golo ser marcado. Este modelo é então aplicado a cada oportunidade de golo durante um jogo, por uma equipa ou atleta, para calcular o valor de golos que seriam expectáveis marcar

na partida pela equipa/atleta em análise. Este valor fornece uma indicação do desempenho geral de finalização da equipa/atleta (Brechot & Flepp, 2020; StatsBomb, 2022b).

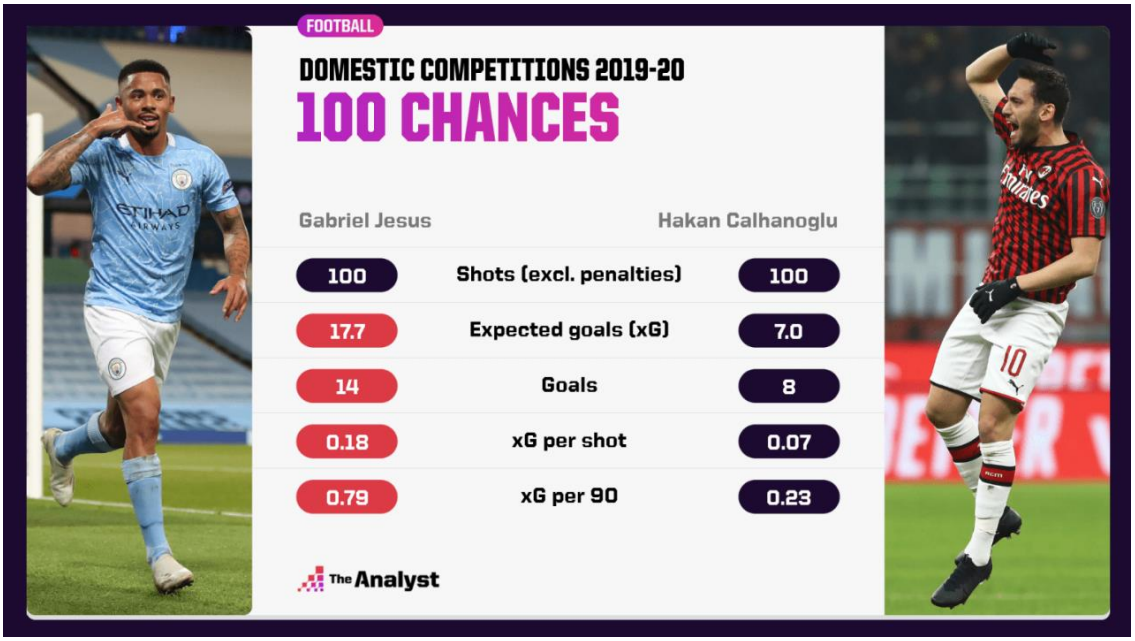


Figura 2: Comparação Estatística de 100 Chances de Golo de Gabriel Jesus e Hakan Çalhanoglu (Whitmore, 2021c)

Na figura 2 é possível comparar as 100 chances dos jogadores Gabriel Jesus, do Manchester City, e do Hakan Çalhanoglu, do Milan, na época 2019-20. Através dos valores de xG é possível perceber que as chances de golo do jogador do City, são substancialmente melhores, pois era espectável que houvesse cerca de 18 golos das 100 chances criadas, em comparação com os apenas 7 golos de Çalhanoglu. No entanto, este apenas marcou 14 dessas chances, o que revela que o jogador esteve abaixo das expectativas, *underperforming*, e o seu oponente, em *overperforming*, por conseguir marcar mais do que o esperado.

Existem diversos fatores a considerar no cálculo e este pode variar consoante os autores (Whitmore, 2021c, 2022). A versão inicial do modelo da empresa Stats Perform, por exemplo, era um modelo de regressão logística e foi treinada com dados provenientes da empresa Opta (Whitmore, 2021c). Esta versão considerava variáveis como a distância ao golo, o ângulo de remate, a parte do corpo de remate e a situação de jogo, ou seja, se a chance vinha de um canto, livre direto ou uma jogada corrida.

O modelo atual é uma rede neuronal artificial que já tem em consideração mais variáveis relacionadas com o posicionamento dos defesas adversários e do guarda-redes (Whitmore,

2022). A pressão exercida pelos defesas, quanta visibilidade tem o avançado da baliza, a posição do guarda-redes, maior granularidade de tipos de remate, como remate de cabeça ou vólei e mais detalhes de contexto, isto é, se a chance provém de um ressalto, ou se é um remate ao primeiro toque, são algumas das melhorias feitas neste novo modelo (Whitmore, 2021c, 2022).

Figura 3: Melhoria no xG da versão atual do modelo da Stats Perform (Whitmore, 2022)

Esta melhoria permitiu refletir um valor de xG mais próximo da realidade para alguns tipos de chances, como as mostradas acima.

É relevante ressaltar que existem condições especiais em que o cálculo difere. Os Penaltis e as recargas são esses casos. Para o primeiro caso, a taxa de conversão em golo é de 75% e portanto a maioria dos autores atribui o valor de 0.75 ao xG do tipo Penalti (The Punters Page, 2023). Em relação às recargas, a fórmula pode ser traduzida como:

$$xG = (1 - (1 - xG_1) + (1 - xG_2)) \quad (1)$$

No caso das recargas, o xG não deve ser calculado como a soma de ambas as situações, mas sim como a probabilidade de não ter sido golo de ambos os lances, subtraindo as probabilidades de ambos por 1 (The xG Philosophy, 2020).

Os xG são uma ferramenta útil para analisar jogos de futebol porque fornecem uma visão mais detalhada do desempenho de uma equipa e têm, simplesmente, em conta o número de golos marcados. Isso pode ajudar a identificar áreas em que uma equipa está a criar oportunidades de golo de alta qualidade e onde pode estar a ter dificuldades em fazê-lo (Brechot & Flepp, 2020; Whitmore, 2021c).

2.3.2 xT

No futebol, há jogadores como Cristiano Ronaldo, Lionel Messi ou Neymar Júnior cujo impacto no resultado de um jogo pode ser visto nos seus números estatísticos (M. Van Roy et al., 2020). No entanto, alguns jogadores podem ser desvalorizados pelas estatísticas (Singh, 2019). Durante anos, a falta de um sistema para recompensar os jogadores envolvidos na formação das jogadas, frustrava os analistas de dados (Tripathy, 2022).

Pegando num exemplo prático, um médio-centro criativo estava dependente do golo do avançado, para que este recebesse o prémio estatístico, a assistência. Em caso de falha, o jogador que fez o passe, não teria qualquer recompensa estatística. Em suma, a eficiência do atacante determinava as estatísticas dele e do seu companheiro de equipa. Isto aconteceu até que surgiram novas métricas que proporcionaram uma melhor compreensão e mais profunda do jogo. Métricas como a ameaça esperada, "*expected threat*" (xT), a assistência esperada (xA) e o valor ofensivo esperado adicionado (xOVA), que irá ser abordado mais à frente neste capítulo, ajudaram os jogadores menos finalizadores a receber a sua devida parte (Everett et al., 2022; Pulis & Bajada, 2022).

Expected Threat é uma estatística criada por Karun Singh, que segue o modelo de Markov, com o objetivo de medir a probabilidade de uma ação de ataque de uma equipa terminar num remate à baliza. O modelo desenvolvido pelo autor atribui pontuações individuais às ações da construção de jogo, ignorando o que aconteceu antes ou irá acontecer nas ações seguintes. Usa dados de eventos, como o local de início e fim da ação e qual o jogador com a posse de bola. Para atribuir pontuações com base nos locais de início e fim, o autor utiliza um valor atribuído a cada local no campo e calcula a pontuação como o valor do local final menos o valor do local inicial. Inclui também no cálculo o uso de xG e são reconhecidas posições mais "ameaçadoras" que permitem a possibilidade de encadear múltiplas ações e gerar altos níveis de ameaça (Singh, 2019). Assumindo que se divide um campo de futebol numa grelha de j colunas por i linhas, a ameaça esperada para a posição (x, y) pode ser traduzida na seguinte fórmula:

$$xT_{x,y} = \left(s_{x,y} \times g_{x,y}\right) + \left(m_{x,y} \times \sum_{z=1}^{j} \sum_{w=1}^{i} T_{(x,y)\rightarrow(z,w)} \times xT_{z,w}\right) \quad (2)$$

em que $s_{x,y}$ corresponde à probabilidade da ação ser um remate e $g_{x,y}$ é a probabilidade do remate ser golo. A multiplicação das duas variáveis anteriores é somada a $m_{x,y}$, que é a probabilidade de a ação ser o movimento da bola para outra área da grelha, isto inclui ação de

passe ou drible. $\sum_{z=1}^{j}\sum_{w=1}^{i} T_{(x,y)\rightarrow(z,w)} \times xT_{z,w}$ corresponde ao somatório da probabilidade de mover a bola para qualquer zona vezes a ameaça esperada dessa zona. É percetível que esta fórmula é recursiva e o cálculo de xT de uma zona necessita do cálculo das restantes zonas. Como tal, a primeira iteração deve ser feita assumindo que xT de todas as zonas corresponde a 0 e desta forma, embora não corresponda exatamente à métrica xG, a primeira iteração representa o quão boa é a posição para um remate à baliza. Durante as seguintes iterações, é avaliado o novo xT para cada zona, usando os valores xT das iterações anteriores. Na segunda iteração, por exemplo, o objetivo é mover a bola para uma zona e, de seguida, rematar. A mesma lógica é estendida para n iterações, ou seja, $xT_{x,y}$ na iteração n, representa a probabilidade de marcar nas próximas n ações (Singh, 2019).

Esta métrica revolucionou o mundo da análise de dados futebolísticos e as suas aplicações são imensas, sendo algumas delas:

- Identificar e analisar padrões de jogo como contra-ataques, através da observação da mudança de xT;
- Avaliar a tomada de decisão de um jogador, de acordo com a maneira de como a sua equipa joga;
- Identificar áreas do campo em que determinada equipa é mais perigosa (Singh, 2019).

Sucintamente, o uso desta métrica permite a equipas e atletas melhorar o seu desempenho e obter vantagem competitiva, observando quais os pontos fortes e fracos dos seus adversários.

2.3.3 VAEP

VAEP ou "*valuing actions by estimating probabilities*" é uma *framework* para avaliar as ações de um jogador de futebol durante uma partida. Introduzido, inicialmente, no *paper* "*Actions Speak Louder than goals*", o método envolve calcular a mudança nas probabilidades de marcar e sofrer para uma equipa como resultado de uma ação especifica, levando em consideração o estado atual do jogo (M. Van Roy et al., 2020).

Assumindo que o estado do jogo é dado como $S_i = [a_1, \dots, a_i]$, a mudança na probabilidade da equipa x marcar, é calculada como a diferença entre a probabilidade de pontuação no estado atual do jogo e a probabilidade de pontuação no estado de jogo anterior:

$$\Delta P_{marcar}(a_i, x) = P_{marcar}(S_i, x) - P_{marcar}(S_{i-1}, x) \quad (3)$$

De igual forma, a mudança na probabilidade da equipa x sofrer é calculada como a diferença nas probabilidades de sofrer no estado atual e no passado:

$$\Delta P_{sofrer}(a_i, x) = P_{sofrer}(S_i, x) - P_{sofrer}(S_{i-1}, x) \quad (4)$$

Este método permite avaliar as ações em termos de impacto nas chances de uma equipa marcar e sofrer num futuro próximo. Tendo em mente que todas as ações têm o objetivo de aumentar a chance de marcar e reduzir a chance de sofrer, o VAEP pode ser dado por:

$$VAEP(a_i, x) = \Delta P_{marcar}(a_i, x) + (-\Delta P_{sofrer}(a_i, x)) \quad (5)$$

Para estimar as probabilidades, os autores referem que qualquer modelo de previsão probabilística pode ser usado (Decroos et al., 2019).

Se os valores das ações individuais de um jogador forem agregados para diferentes granularidades de tempo, é possível ter uma visão da prestação do atleta nesse mesmo período.

Comparativamente às métricas tradicionais como golos por 90 minutos, assistências por 90 minutos ou golos + assistências por 90 minutos, a *framework* aqui descrita consegue traduzir um valor mais real dos atletas e precisar melhor o desempenho do jogador do que as métricas tradicionais. A tabela 2 reflete esta comparação.

Tabela 2: Lista dos jogadores mais valiosos da liga inglesa na época 2017-18 de acordo com VAEP (Decroos et al., 2019)

R_{vaep}	Player	Rating	R_g	R_a	R_{g+a}	Market Value
1	P. Coutinho	0.899	10	2	4	€ 140m
2	M. Salah	0.817	1	23	2	€ 150m
3	K. De Bruyne	0.641	72	4	15	€ 150m
4	E. Hazard	0.636	21	122	34	€ 150m
5	R. Mahrez	0.635	34	11	16	€ 60m
6	A. Martial	0.607	13	13	9	€ 60m
7	R. Sterling	0.579	7	6	5	€ 120m
8	P. Pogba	0.549	55	9	28	€ 80m
9	H. Kane	0.545	4	140	6	€ 150m
10	S. Heung-Min	0.539	19	36	17	€ 50m

Outro caso de uso desta métrica poderia ser a aquisição de jogadores jovens e promissores, apresentado na tabela 3.

Tabela 3: Lista dos Jogadores jovens mais promissores na época 2017-18 ordenados por VAEP (Decroos et al., 2019)

Rank	Name	Team	Age	Rating	Market Value
1	D. Neres	Ajax	21	0.620	€ 25m
2	M. Mount	Vitesse	19	0.616	€ 4m
3	Malcom	Bordeaux	21	0.567	€ 40m
4	K. Mbappé	PSG	19	0.507	€ 200m
5	F. de Jong	Ajax	20	0.495	€ 60m

Relativamente à já apresentada métrica xT, ambas as abordagens visam avaliar como as ações aumentam ou diminuem a probabilidade de produzir um golo. xT tem uma representação limitada do estado do jogo que é puramente baseado na localização, enquanto o VAEP tem uma representação mais detalhada baseada em recursos que captura a ação e o contexto de jogo. A figura 4 apresenta algumas diferenças entre estas abordagens (M. Van Roy et al., 2020).

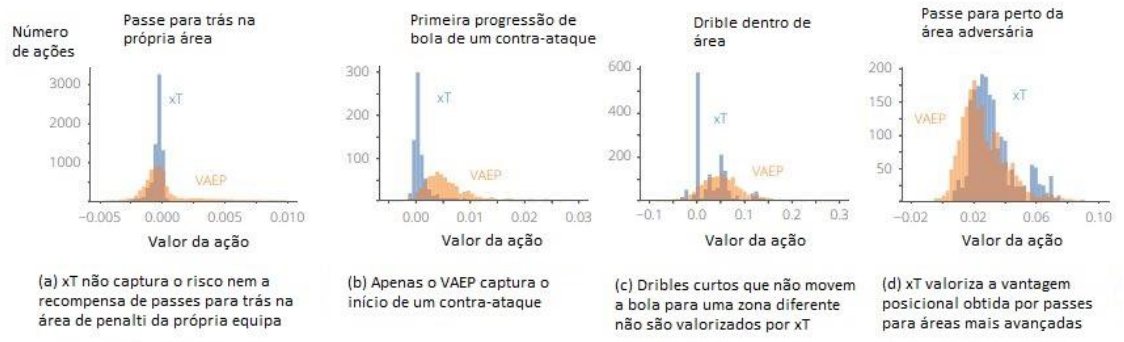


Figura 4: Histogramas dos valores de xT e VAEP para um conjunto de ações (M. Van Roy et al., 2020)

A framework VAEP fornece uma abordagem simples de avaliar ações que é independente da representação usada para descrever as ações. O seu ponto forte relativo a outras abordagens, é que esta transforma a tarefa subjetiva de avaliar uma ação na tarefa objetiva de prever a probabilidade de um evento futuro de maneira natural.

2.3.4 xA

xA é uma abreviação para "*expected Assist*" ou assistência esperada (Whitmore, 2021b). Uma assistência é o último passe para o autor do golo. xA mede a probabilidade de um determinado passe se tornar um passe que leve a um golo e como tal, está intimamente ligada á métrica xG,

já mencionada. xA é tão maior quanto o passe for dado para um local onde o xG é elevado (Goes, 2021; Worville, 2022). Para tal, são considerados fatores como (Whitmore, 2021b):

- a localização de onde o passe é feito, a localização final do passe e a sua distância;
- o tipo de passe, se é um cruzamento, passe rasteiro, cabeceamento ou lançamento com a mão;
- momento de jogo, como um canto, cobrança de falta ou reposicionamento lateral ou jogada normal.

A soma das assistências esperadas de um jogador indica uma previsão de quantas assistências um jogador fará. xA é uma indicação melhor da qualidade de passe de um jogador e de quão bom um jogador é a fazer assistências do que ter em consideração o real número de assistências. Olhando para um exemplo prático, na figura abaixo estão presentes os números relativos a assistências, para os dois defesas laterais titulares do Liverpool na época 2019-20, Andrew Robertson e Alexander-Arnold:

Figura 5: Comparação Estatística de 41 Chances Criadas por Andrew Robertson e Trent Alexander-Arnold (Whitmore, 2021b)

Ambos os jogadores têm o mesmo número de chances e para tal, eram esperadas cerca de 5 assistências para Robertson, no entanto, este excedeu as espectativas, totalizando 10 assistências. Por outro lado, Arnold realizou 6 assistências, enquanto eram esperadas 7. Isto deveu-se ao facto de algumas assistências realizadas por Andrew Robertson serem realizadas para locais onde o avançado ainda teria um xG baixo, apesar dos cruzamentos de Arnold terem

maior qualidade, mas o avançado não ter conseguido converter em golo. As figuras que ilustram esta justificação, correspondem às imagens 70 e 71 deste documento e estão presentes em anexo.

Métricas como a aqui referida tornam-se mais prevalentes no futebol nos últimos anos e atletas e treinadores falam abertamente sobre os mesmos. O diretor desportivo do Norwich City, Stuart Webber, revelou que usaram a xA no recrutamento do seu melhor jogador nas últimas temporadas, Emiliano Buendia. Embora os seus números, na segunda divisão espanhola, não fossem substancialmente bons, estes não traduziam o seu real valor e a métrica xA foi utilizada para reconhecer a criatividade subjacente ao jogador (Whitmore, 2021b).

2.3.5 xOVA

O valor ofensivo esperado adicionado, "*expected offensive value added*" (xOVA) é uma estatística avançada usada na análise de futebol que mede o valor que um jogador agrega ao total de golos esperados da sua equipa, através das suas ações ofensivas (Tripathy, 2022). Existem diversas formas de calcular o xOVA. Uma versão do cálculo envolve o nível médio da liga e no caso de um xOVA positivo significaria que um jogador contribuiu com mais valor para o total de golos da sua equipa do que a média da liga, enquanto o negativo indicaria que que contribuiu com menos valor (Soccerment, 2022a, 2022b). A versão desenvolvida pelo site Soccerment é mais simples (Tripathy, 2022):

$$\text{xOVA} = (\text{xG chance sem ser penálti} + \text{xA}) - \text{xA recebido} \quad (6)$$

É adicionado ao xG a assistência espectável. De seguida, é subtraído a xA recebida dos companheiros de equipa. Isto demonstra o valor ofensivo que o jogador traz à equipa sem o valor recebido dos companheiros. Também é possível perceber que esta versão tem em consideração a qualidade do remate ou do passe, e não têm em conta os seus resultados, pois apenas contêm valores esperados. Assim, um jogador recebe a sua quota parte do valor pela jogada, independentemente do facto do colega de equipa marcar ou falhar (Tripathy, 2022).

O xOVA pode ser uma métrica útil para ajudar a identificar jogadores com desempenho superior ou inferior em relação à média da liga em termos de contribuição ofensiva. Também pode ser usado para avaliar o impacto individual de um jogador no ataque da equipa desempenhando diferentes posições (Soccerment, 2022b).

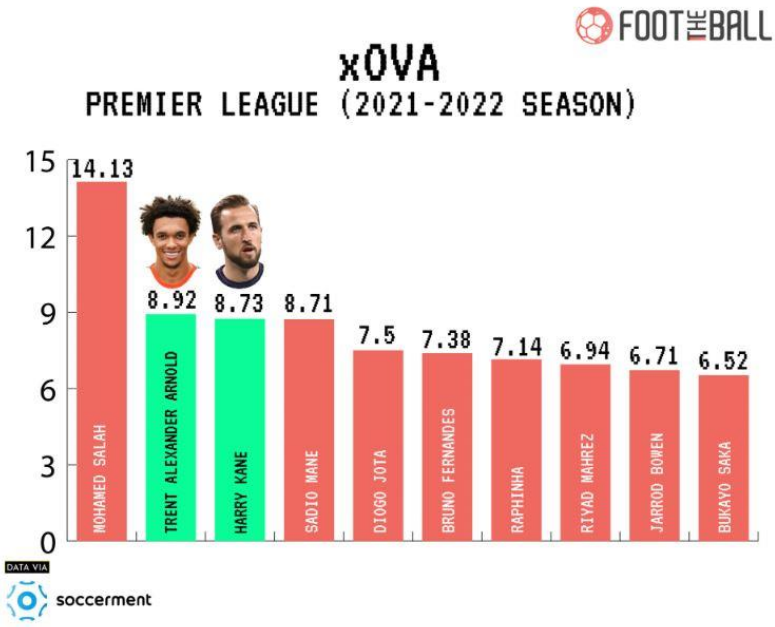


Figura 6: 10 melhores jogadores da liga inglesa, em termos de xOVA, para a época 2021-2022 (Tripathy, 2022)

Na figura 6 é feita uma comparação entre os 10 melhores jogadores, em termos de xOVA, para a época 2021-2022, na liga inglesa. Tomando os jogadores com as barras verdes como exemplo, Alexander-Arnold do Liverpool e Harry Kane do Tottenham, e aplicando a fórmula 6:

xOVA de Alexander-Arnold = (1.80 + 12.08) – 4.96 = 8.92
xOVA de Harry Kane = (14.20 + 5.13) – 10.6 = 8.73

É possível verificar que apesar do Kane ser avançado, contribuir com bastantes golos e ter um xG elevado, os companheiros de equipa sempre vão tentar passar-lhe a bola para marcar e consequentemente a sua xA recebida também irá ser elevada. Em contraste, o valor de xOVA do Arnold sobrepõem-se pois é um defesa, que raramente marca golos, os seus colegas não tentam fazer-lhe assistências, mas contribui ofensivamente, ele próprio, com assistências e daí o seu elevado xA. Este é um exemplo que torna o xOVA, uma métrica tão eficaz.

Esta métrica é maioritariamente usada em conjunto com outras estatísticas e análises com o objetivo de obter uma compreensão mais abrangente e completa do desempenho de um atleta.

Após o esclarecimento acerca das métricas apresentadas, as questões Q-02 e Q-03 foram respondidas.

2.4 Posse vs. Sequência

Após o leitor estar familiarizado com as métricas avançadas relacionadas com o futebol, é necessário apresentar os conceitos de posse e sequência. Estes conceitos são o *core* da análise avançada no futebol e importantíssimos no futebol moderno (Whitmore, 2021a).

No futebol, as sequências são definidas como passagens de jogo que pertencem a uma equipa e são encerradas por ações defensivas, interrupções no jogo ou um remate (Harkins, 2022; Whitmore, 2021a). Uma sequência tem início quando um jogador realiza uma ação controlada com a bola. Isto inclui passes ou dribles, mas não ações defensivas, como cortes ou interceções, a menos que sejam seguidos por ações controladas (Harkins, 2022; Whitmore, 2021a).

Por outro lado, uma posse pode ser definida como uma ou mais sequências seguidas pertencentes à mesma equipa. Uma posse é, também encerrada, quando a equipa adversária, ganha o controlo da bola (Harkins, 2022; Whitmore, 2021a). Uma série de passes que leva a um remate que é defendido e resulta num canto, constituiria uma posse de bola, mas duas sequências. A sequência original e a sequência iniciada a partir do canto seriam ambas incluídas na mesma posse de bola (Harkins, 2022; Whitmore, 2021a).

Com isto em mente, existem algumas suposições a ter em mente:

- O número total de posses de cada equipa, apenas difere em um. Ou seja, se a posse de bola de uma equipa termina, a da equipa adversária começa;
- A duração dessas mesmas posses, não é necessariamente igual, em termos de número de sequências e de tempo;
- Existem eventos que não pertencem a nenhuma sequência ou posse, por exemplo a defesa do guarda-redes, do exemplo apresentado em cima, porque a ação seguinte não foi controlada e, portanto, não deu inicio a nenhuma sequência.

Atribuindo particular foco às sequências, pois são um dos temas centrais deste projeto, estas podem fornecer métricas interessantes sobre jogadores e equipas.

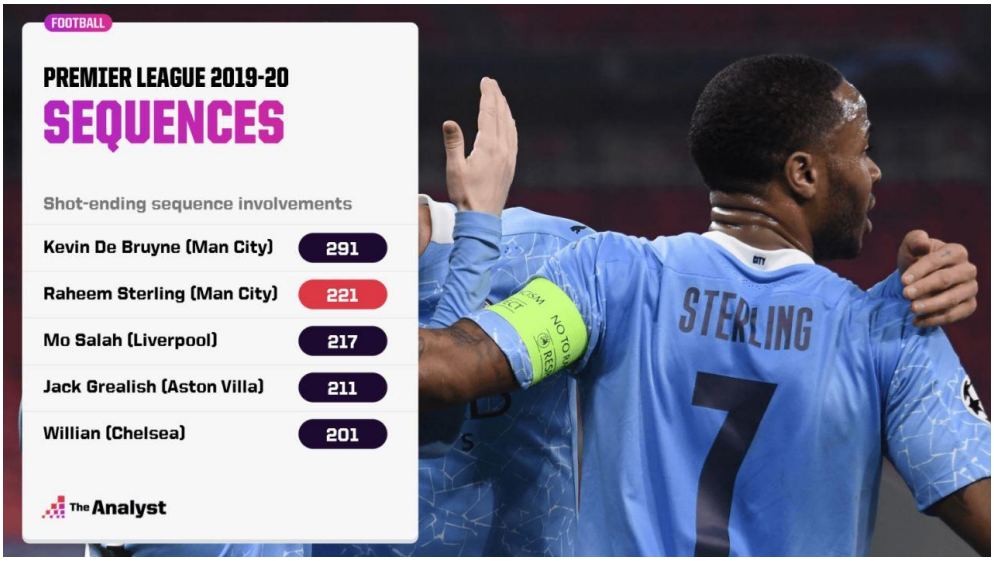


Figura 7: 5 Jogadores que mais participam de Sequências que terminam em remate na liga Inglesa na Época 2019-20 (Whitmore, 2021a)

Na figura 7 é possível verificar quais os 5 jogadores mais envolvidos em sequências que terminam com um remate à baliza, ou na figura a baixo, é possível perceber o tipo de sequências, preferencial de cada equipa da Liga Inglesa. O Manchester City, opta por sequências mais lentas e com um elevado número de passes, enquanto o West Ham aposta em jogadas mais rápidas com o menor número de passes possível.

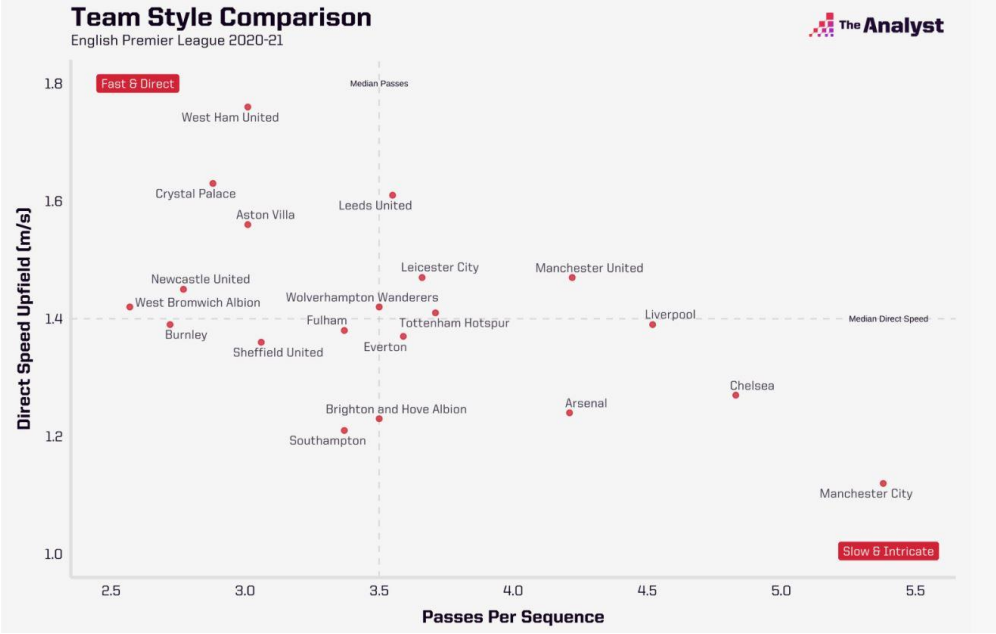


Figura 8: Comparação do Estilo de Sequências das Equipas da Liga Inglesa na Época 2022-2021 (Whitmore, 2021a)

A partir da análise das sequências é possível perceber padrões e classificar as mesmas de acordo com estes mesmos padrões. Adicionalmente, sequências complexas podem pertencer percentualmente a mais que um tipo. Os seguintes tipos de sequências são os mais comuns (Stats LLC, 2022):

- Manutenção da Posse de Bola: troca de passes dentro do meio-campo defensivo;
- Construção de Jogo: sequência de passes na primeira metade do meio-campo ofensivo em que a equipa procura oportunidades para atacar;
- Ameaça Contínua: posse de bola no ultimo terço do campo. O tempo de posse deve ser superior a 6 segundos para ser considerado deste tipo;
- *Fast Tempo*: sequência de alto ritmo com o objetivo de aumentar a velocidade de jogo. Deve ocorrer no meio-campo adversário e pode ser conseguida quando o jogador passa a bola para um companheiro de equipa em menos de 2 segundos ou quando dribla em ritmo acelerado;
- Jogada Direta: instância de jogo em que a equipa tenta mover a bola rapidamente na direção da baliza adversária, através de passes longos. Concretamente, é analisada a distância ganha através de um evento de passe longo, pontapé de baliza ou lançamento do guarda-redes. A distância deve ser superior a 20 metros para ser considerada deste tipo;
- Contra-ataque: ocorre quando uma equipa recupera a bola e tenta, rapidamente, chegar à baliza adversária. A velocidade de transição para o local de destino, determina o valor do contra-ataque. Quanto mais rápido, maior será o valor;
- Cruzamento: consiste num lançamento da bola de uma área ampla do campo com a intenção de encontrar um companheiro de equipa;
- Pressão Alta: indica o quão alto no terreno a equipa recupera a posse de bola. Existem dois fatores a considerar. O primeiro tem a ver com o local da recuperação e o segundo com o tempo que a equipa adversária teve a bola. Este segundo ponto prende-se pelo facto de desconsiderar recuperações de duelos de jogo e capturar os esforços por uma pressão controlada. Deve ser recuperada 15 metros depois do meio-campo e a equipa adversária devia ter a posse de bola por pelo menos 10 segundos.

Observar o jogo através de sequências permite uma análise mais eficaz dos jogadores e das equipas e oferece um nível de perceção que ajuda na compreensão mais informada das funções

dos jogadores e dos estilos de jogo das equipas (Whitmore, 2021a). Depois disto, é possível afirmar que as Q-04 e Q-05 foram respondidas.

2.5 Machine Learning

Depois de apresentados todos os conceitos específicos relacionados com o futebol, é necessário apresentar os conceitos gerais relacionados com aprendizagem automática, aprendizagem profunda e com os modelos mais usados no cálculo das métricas apresentadas no subcapítulo 2.3.

Machine Learning (ML) ou em português Aprendizagem Automática é um subcampo da Inteligência Artificial que lida com o design e desenvolvimento de algoritmos que permitem que os computadores aprendam com os dados e tomem decisões com base nessa aprendizagem. Esses algoritmos são capazes de aprender e melhorar com o tempo, sem a necessidade de programação explícita (Mahesh, 2018; Zhou, 2021).

Existem vários tipos de ML, incluindo aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço.

A aprendizagem supervisionada envolve o treino de um modelo, a partir de um conjunto de dados rotulados, onde a saída correta é fornecida para cada exemplo, no conjunto de dados. Isso permite que o modelo aprenda a relação entre os dados de entrada e a saída correta e faça previsões sobre novos dados, não vistos pelo modelo. Exemplos de aprendizagem supervisionada incluem prever se um cliente irá desistir ou não da subscrição de uma licença, com base no seu comportamento anterior, ou identificar o tipo de objeto numa imagem, com base num conjunto de dados rotulados de imagens (Mahesh, 2018).

Aprendizagem não supervisionada implica o treino de um modelo num conjunto de dados sem rótulos, no qual o modelo deve descobrir a estrutura subjacente dos dados atendendo a padrões e correlações. Exemplos deste tipo incluem agrupar pontos de dados em grupos ou identificar anomalias nos dados (Mahesh, 2018).

O último tipo, aprendizagem por reforço, envolve treinar um modelo para tomar decisões num ambiente dinâmico e em constante mutação, onde o modelo recebe recompensas ou punições com base nas suas ações. Este tipo de aprendizagem é frequentemente utilizado em robótica e

jogos de vídeo para permitir que o agente (máquina ou software) aprenda por tentativa e erro (Mahesh, 2018).

A aprendizagem automática tem uma ampla gama de aplicações, incluindo o reconhecimento de imagem e fala, processamento de linguagem natural, deteções de fraude e carros autónomos. Tem o potencial de revolucionar muitos setores e já teve um impacto significativo na forma como o ser humano interage com a tecnologia. No entanto, também há considerações éticas a serem levadas em consideração no uso destas tecnologias, como problemas de *bias* nos dados e o seu uso para o mal. É importante que pesquisadores e profissionais considerem cuidadosamente os potenciais impactos do seu trabalho e trabalhem para criar sistemas justos e responsáveis (Zhou, 2021).

No que toca ao futebol, a IA está a ser utilizada para aprimorar vários aspetos da indústria, como rastreamento das posições dos jogadores durante as transmissões, automatização do conteúdo dos jogos e identificação de futuras estrelas. Para além disto, o uso de Aprendizagem Automática está impactando o desempenho das equipas pela análise de desempenho, recrutamento de jogadores e planeamento estratégico de longo prazo (Stats Perform, 2022a).

No geral, ML é um campo em rápida evolução que tem o potencial de transformar muitos aspetos das nossas vidas. É uma área empolgante de pesquisa e desenvolvimento, com infinitas possibilidades de inovação e aplicação, futebol incluído.

2.6 Deep Learning

Coloquialmente é frequente pensar em "*Deep Learning*", ou Aprendizagem Profunda, como algo à parte de ML, no entanto esta corresponde a tipos de Redes Neuronais Artificiais que são um tipo de *Machine Learning* que é uma das áreas da Inteligência Artificial, como é possível ver na figura abaixo.

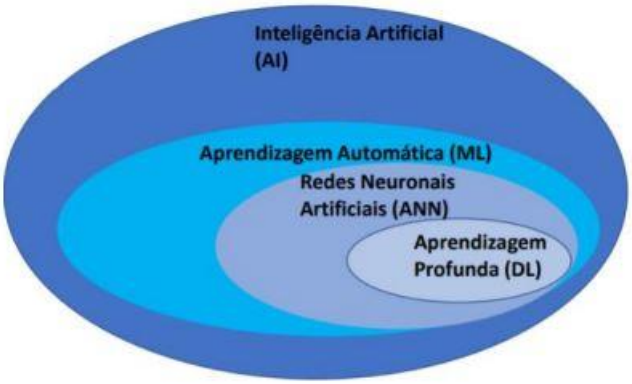


Figura 9: Camadas da Inteligência Artificial (Ramos, 2020)

DL envolve o uso de redes neuronais que são modeladas de acordo com a estrutura e função do cérebro humano.

Sucintamente, as redes neuronais artificiais consistem em camadas de neurónios interconectados, que processam e transmitem informações. Cada neurónio recebe a entrada de outros neurónios e usa essa entrada para realizar uma computação, antes de transmitir o resultado para outros neurónios na próxima camada.

Os algoritmos de Aprendizagem Profunda são capazes de aprender e melhorar com o tempo, sem a necessidade de programação explícita, ajustando os pesos e *biases* das conexões entre os neurónios. Esse processo é conhecido como treino de uma rede neuronal (Kelleher, 2019).

Uma das principais diferenças entre ML e DL é o nível de intervenção humana necessária. No ML tradicional, as características dos dados devem ser extraídas manualmente e escolhidos pelo cientista de dados. Em contraste, os algoritmos de Aprendizagem Profunda são capazes de aprender características automaticamente a partir dos dados brutos, sem a necessidade de "*feature engineering*" manual. Isto torna DL particularmente adequado para tarefas como reconhecimento de imagem e fala, em que os dados brutos são de alta dimensão e complexidade (Chollet, 2021).

Outra diferença é a quantidade de dados necessária para treinar um modelo. Os algoritmos de DL geralmente exigem conjuntos de dados muito maiores do que os algoritmos tradicionais de ML, a fim de aprender os intrincados padrões e relacionamentos nos dados.

Aprendizagem Profunda teve um impacto significativo em muitos campos, incluindo visão computacional, processamento de linguagem natural e saúde. Esta área tem o potencial de

revolucionar a maneira como o ser humano interage e entende dados complexos e já foi usado para obter resultados de ponta numa ampla gama de tarefas. No entanto, também existem limitações para o DL. Pode ser computacionalmente intensivo e requer hardware especializado, como unidades de processamento gráfico (GPUs), para treinar redes neuronais complexas. Além disso, os algoritmos de DL podem ser suscetíveis ao *overfitting,* que ocorre quando o modelo é muito complexo e aprende o ruído nos dados, e não nos padrões subjacentes. Isso pode levar a uma má generalização para dados novos e nunca vistos (Chen & Ran, 2019).

Em resumo, DL é um subcampo de ML que envolve o uso de redes neuronais artificiais para aprender com os dados. É adequado para tarefas como reconhecimento de imagem e fala e requer grandes quantidades de dados para treino. Embora tenha alcançado resultados impressionantes em muitas áreas, também tem as suas limitações e requer considerações cuidadosas quando aplicado a problemas do mundo real.

2.6.1 Aprendizagem Supervisionada

No contexto do futebol, a aprendizagem Supervisionada pode ser usada numa variedade de tarefas. A maioria dos modelos referidos, usados para cálculo das métricas avançadas, eram modelos de aprendizagem supervisionada. De entre os diversos tipos de modelos destacam-se as redes neuronais artificiais, por ser o mais usado.

2.6.1.1 Redes Neuronais Artificiais

Uma rede neuronal artificial (RNA) é um tipo de algoritmo de aprendizagem supervisionada modelado de acordo com a estrutura e função do cérebro. É composta por nós (neurónios) interconectados, organizados em camadas (Chollet, 2021).

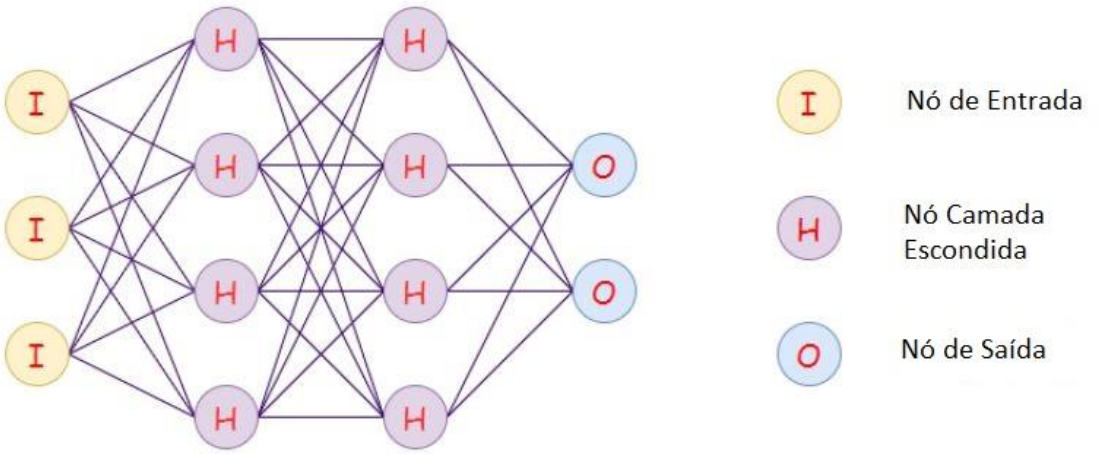


Figura 10: Representação de uma Rede Neuronal Artificial Profunda (Pratik & Iriondo, 2022)

Uma RNA é treinada para realizar uma tarefa específica ajustando os valores das conexões entre os seus neurónios, conhecidas como pesos. Este processo é conhecido como treino da rede neuronal artificial. Para isto ser possível, um grande conjunto de dados é inserido na rede e a saída da rede é comparada com a saída desejada. Com base nessa comparação, os pesos das conexões são ajustados para reduzir o erro entre a saída e a saída desejada. Esse processo é repetido até que o erro seja minimizado e a rede seja capaz de executar a tarefa com precisão. Os próprios nós aprenderão (atualizarão) cada vez que os dados forem propagados e em cada instante a rede representa um *snapshot* do conhecimento atual do sistema (Chollet, 2021; Kelleher, 2019).

Os valores das conexões entre os neurónios são determinados usando uma função matemática chamada função de ativação. A função de ativação determina se um neurónio transmitirá informações ou não com base no valor do sinal de entrada que recebe (Chollet, 2021).

Após o treino de uma RNA, esta pode ser usada para fazer previsões sobre dados novos e nunca vistos, passando os dados de entrada pela rede e gerando uma saída na camada de saída.

2.6.2 Unsupervised learning

O futebol é um desporto apreciado por milhões de pessoas em todo o mundo. Com o advento de tecnologias sofisticadas, tornou-se possível analisar vários aspetos do jogo com maior precisão e descobrir perceções ocultas que, de outra forma, seriam impossíveis de ver. Uma

das ferramentas mais poderosas para esse tipo de análise é a aprendizagem não supervisionada (Cartas et al., 2022; Majumdar et al., 2022; Wijngaard, 2020).

A aprendizagem não supervisionada é uma técnica de aprendizagem automática usada para analisar dados sem a necessidade do seu rótulo. Envolve treinar um modelo num conjunto de dados e, em seguida, usá-lo para identificar padrões e *insights* nos dados que, de outra forma, seriam difíceis de ver (Mahesh, 2018).

No contexto do futebol, a aprendizagem não supervisionada pode ser usada para analisar vários aspetos do jogo, como por exemplo as já mencionadas sequências. Pegando neste exemplo, a aprendizagem não supervisionada pode ser usada para analisar os dados das sequências como posição dos jogadores, movimento e ações, para com isto classificá-las. Isso pode ser feito usando algoritmos de agrupamento ou técnicas de redução de dimensionalidade que identificam padrões nos dados que correspondem a diferentes tipos de ações, posições, formações ou sequências de passes (Cartas et al., 2022; Wijngaard, 2020). Assim, a Q-06 pode ser dada como respondida.

A aprendizagem não supervisionada é uma ferramenta extremamente poderosa para descobrir informações ocultas nos dados do futebol. Permite que analistas obtenham novas perspetivas sobre o jogo e tomem decisões mais bem informadas. Abaixo são apresentados dois exemplos que são usados atualmente no futebol moderno para obter estes *insights*.

2.6.2.1 Autoencoders

Um autoencoder é um tipo de rede neuronal treinada para reconstruir as suas entradas. Consiste em duas partes principais: um codificador e um descodificador. O codificador pega nos dados de entrada e faz o seu mapaemento para uma representação latente de dimensão inferior, conhecida como bottleneck ou representação codificada. O descodificador então mapeia a representação codificada de volta ao espaço de entrada original, produzindo uma reconstrução dos dados originais (A. Roy, 2020).

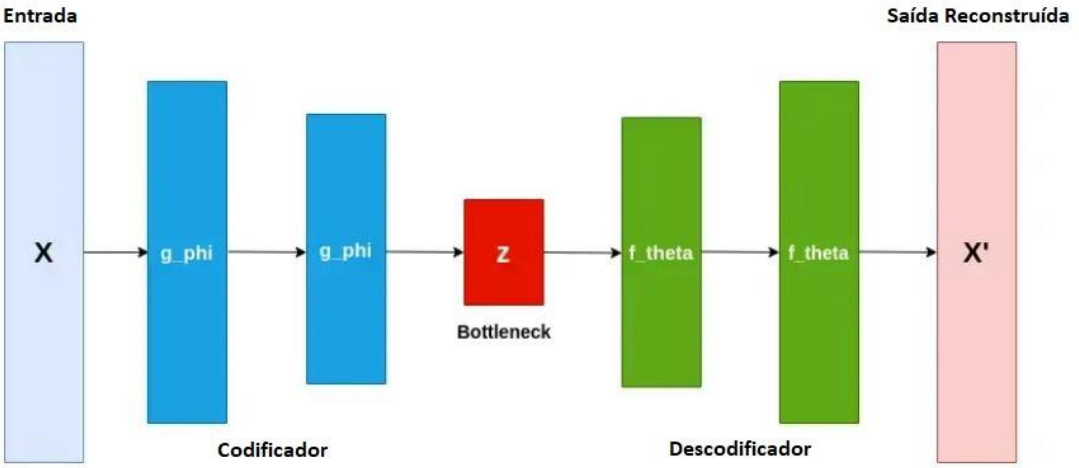


Figura 11: Representação de um Autoencoder (A. Roy, 2020)

O autoencoder é treinado minimizando a diferença entre a entrada original e a reconstrução. Isso é feito minimizando uma função de perda, como o erro quadrático médio entre a entrada original e a reconstrução (MSE). Durante o treino, o codificador aprende uma representação compacta e informativa dos dados de entrada, enquanto o descodificador aprende como reconstruir os dados de entrada a partir da representação codificada (A. Roy, 2020).

Existem diferentes variações de autoencoders, como autoencoders de redução de ruído e autoencoders variacionais (VAE). Os autoencoders de redução de ruído são treinados para reconstruir uma entrada a partir de uma versão ruidosa da entrada, enquanto os autoencoders variacionais são treinados para reconstruir entradas usando latentes estocásticos que são probabilísticos e podem ser vistos como um modelo generativo (A. Roy, 2020).

Uma vez treinado, um autoencoder pode ser usado para tarefas como redução de dimensionalidade, extração de recursos, deteção de anomalias e compactação de imagem. Também pode ser usado como um extrator de recursos para outras redes neuronais. Por exemplo, a representação codificada aprendida pelo autoencoder pode ser usada como entrada para um classificador supervisionado, o que pode melhorar seu desempenho (Pratik & Iriondo, 2022).

2.6.2.2 GANs

As redes Adversariais Generativas, ou GANs do inglês "*Generative Adversarial Networks*", são um tipo de arquitetura de rede neuronal, que envolve o treino de um modelo para gerar novos

dados inéditos que são semelhantes a um determinado conjunto de dados de treino (Pratik & Iriondo, 2022).

Á semelhança dos autoencoders, também são compostas por dois componentes principais: uma rede geradora e uma rede discriminadora. A rede geradora recebe ruído aleatório como entrada e produz novos dados, enquanto a rede discriminadora recebe tanto os dados gerados quanto os dados reais e tenta distingui-los. A rede geradora é treinada para produzir dados indistinguíveis dos dados reais, enquanto a rede discriminadora é treinada para identificar corretamente quais dados são reais e quais são gerados (Alqahtani et al., 2021).

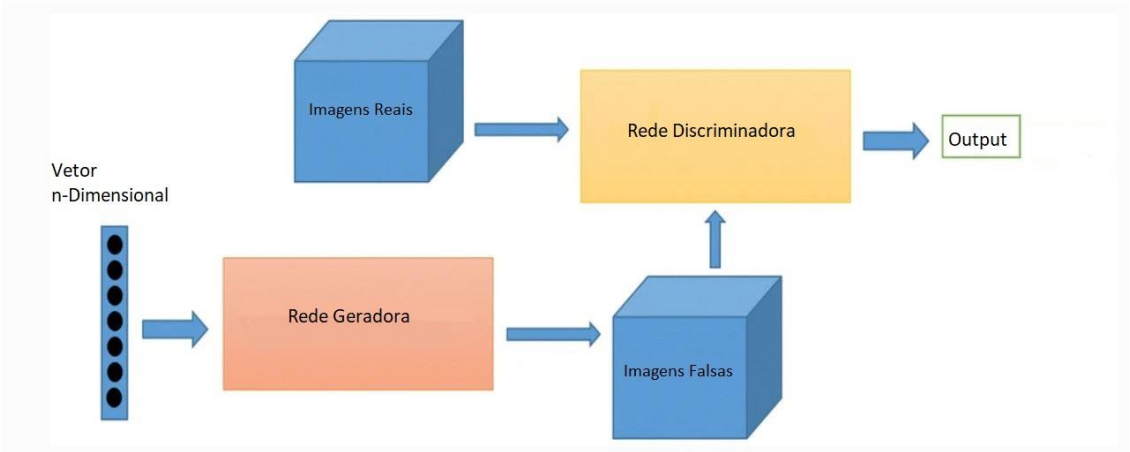


Figura 12: Representação de uma Rede Adversarial Generativa (Alqahtani et al., 2021)

Ambas as redes são treinadas simultaneamente, com a rede geradora a tentar produzir dados que possam enganar a rede discriminadora, e a rede discriminadora a tentar identificar corretamente os dados gerados, ou seja a rede discriminadora é usada como uma avaliação. À medida que o treino avança, a rede geradora torna-se melhor na produção de dados realistas, enquanto a rede discriminadora torna-se melhor na identificação dos dados gerados. A rede geradora é treinada para produzir amostras de dados que correspondam aos dados reais para minimizar a diferença. O resultado final do processo de treino de uma GAN é uma rede geradora que pode produzir novas amostras de dados semelhantes aos dados reais (Alqahtani et al., 2021).

As GANs podem ser usadas para uma ampla gama de tarefas, como síntese de imagem, transferência de estilo e geração de texto, e é uma ferramenta poderosa quando há escassez de dados.

É importante observar que o treinamento de GANs pode ser muito difícil, pois pode ser difícil estabilizar e equilibrar o treino entre as redes, geradora e discriminadora. Esse fenômeno é conhecido como modo de colapso, onde a rede geradora produz uma gama limitada de saídas, por isso é importante escolher a arquitetura certa e ajustar adequadamente os hiperparâmetros para obter os melhores resultados (Alqahtani et al., 2021).

2.7 Soluções Existentes

Uma vez que o leitor está mais familiarizado com esta área devido aos subcapítulos anteriores, neste subcapítulo são expostas abordagens que outros autores apresentaram para a resolução do problema relacionado com o presente projeto. O objetivo é mostrar ao leitor que as ferramentas usadas neste projeto foram também usadas em trabalhos semelhantes, fazendo sentido o uso das mesmas e obter a resposta para a questão Q-07 definida.

2.7.1 SAP-Sports-One

A SAP é uma corporação de software multinacional alemã fundada em 1972 que fabrica software empresarial para gerenciar operações de negócios e relações com clientes. Devido à sua poderosa capacidade de computação na *cloud*, a SAP fornece vários tipos de soluções, incluindo análise de dados e bases de dados. Especificamente, a SAP oferece o SAP-Sports-One (Football Analytics 101, 2019).

O SAP-Sports-One é um conjunto de soluções de software projetado especificamente para a indústria desportiva. Oferece uma gama de ferramentas e recursos destinados a ajudar organizações e equipas desportivas a gerenciar, analisar e otimizar o seu desempenho. Alguns dos principais recursos incluem (SAP Sports One, n.d.):

- Gerenciamento de Equipas e atletas: fornece ferramentas para gerenciar informações de equipas e jogadores, incluindo cronogramas, contratos e dados de desempenho. Também oferece ferramentas para gerenciar a saúde e o bem-estar do atleta, como previsão de lesões e planos de reabilitação.
- Análise de desempenho: o SAP Sports One inclui ferramentas para analisar e visualizar dados de desempenho, como estatísticas de partidas e dados de rastreamento de jogadores. Isto ajuda equipas e treinadores a identificar tendências e padrões e tomar decisões informadas sobre táticas e desenvolvimento de jogadores.

- Análise de Vídeo: oferece ferramentas de análise de vídeo que permitem que as equipas revisem e analisem imagens de vídeo de jogos e sessões de treino. Esta ferramenta pode ser usada para identificar pontos fortes e fracos e para identificar áreas de melhoria.
- Envolvimento dos fãs: inclui ferramentas para o envolvimento dos fãs, incluindo integração de mídia social e soluções de bilheteira.

No geral, o SAP-Sports-One foi projetado para ajudar organizações e equipas desportivas a simplificar as suas operações, melhorar o desempenho e aprimorar a conexão com os adeptos. É usada por uma ampla gama de equipas e organizações desportivas, tendo até sido essencial na campanha da seleção alemã, no mundial de 2014, onde esta foi campeã mundial (SAP Sports One, n.d.).

2.7.2 InStat Scout

O InStat Scout é uma plataforma de análise de futebol que fornece uma variedade de ferramentas e recursos para analisar e visualizar dados de futebol. É projetado para ajudar treinadores, analistas e outras partes interessadas a obter informações sobre o desempenho da equipa e dos atletas e a tomar decisões informadas sobre táticas e desenvolvimento do jogador. Foi criada pela empresa InStat, empresa especializada na análise de desempenho desportivo fundada em Moscovo, na Rússia, em 2007 (Football Analytics 101, 2019). Os principais recursos disponibilizados por esta plataforma incluem: visualização de dados através de representações gráficas, ferramentas de análise de desempenho, análise de vídeo e gerenciamento de equipas, semelhante à solução apresentada anteriormente. A referida visualização de dados inclui a representação gráfica de métricas avançadas como as apresentadas no subcapítulo 2.3 e mapas de calor (InStat Scout, n.d.).

Esta solução é uma ferramenta poderosa para análise de desempenho futebolístico e atualmente é bastante usada por vários clubes de futebol, em todo o mundo (InStat Scout, n.d.).

2.7.3 Opta

A Opta é uma empresa de análise desportiva com sede no Reino Unido, fundada em 1996. É uma subsidiária integral do Perform Group desde 2013. A Opta fornece dados para 30 desportos em 70 países, atendendo a uma ampla gama de clientes, incluindo ligas, emissoras e sites de

apostas. Além do futebol, a Opta fornece dados detalhados para uma variedade de outros desportos. A empresa coleta dados de eventos e rastreamento e vende-os para empresas e indivíduos (Football Analytics 101, 2019). Todos os anos, a Opta realiza o Sports Analytics Forum, onde analistas de futebol apresentam as suas pesquisas e descobertas. Os vídeos do Fórum OptaPro Analytics 2019 podem ser acessados online e oferecem informações sobre os últimos desenvolvimentos na área (Stats Perform, 2022b, 2022c).

2.7.4 StatsBomb IQ

StatsBomb é uma empresa de dados desportivos que foi criada por analistas para analistas e tem uma equipa crescente dedicada a coletar e analisar dados desportivos abrangentes de todo o mundo. A plataforma da empresa foi projetada para permitir a coleta e análise de dados mais relevantes do que qualquer outra solução concorrente e ser flexível o suficiente para se adaptar a novas necessidades, oportunidades e desafios (Football Analytics 101, 2019).

A StatsBomb começou como um blog de análise de futebol em 2013, fundado por Ted Knutson como um *hub* central para boas análises de toda a Internet. O *blog* rapidamente ganhou reputação de conteúdo orientado a dados de alta qualidade e ajudou a reunir uma comunidade de analistas de todo o mundo. À medida que a empresa ganhava um perfil maior no mundo da análise de futebol, começou a ser abordada por equipas que procuravam assistência com os seus desafios de dados. A StatsBomb começou como uma consultoria, trabalhando para estabelecer uma cultura baseada em dados em organizações de ligas e competições em todo o mundo (StatsBomb, n.d.-b).

Em 2019, a StatsBomb uniu forças com a Arqam FC, uma empresa de análises com sede no Cairo, Egito. O Arqam FC desenvolveu uma maneira de coletar e analisar partidas de futebol detalhadamente, tornando-o um parceiro ideal. Além dos seus negócios analíticos, a Arqam também opera uma agência de criação de conteúdo de mídia, Arqam Digital, que trabalha com clubes e federações, principalmente na região do Médio Oriente e Norte de África (StatsBomb, n.d.-b).

O StatsBomb reconheceu que os dados disponíveis na época tinham lacunas que comprometiam a capacidade dos analistas de obter uma imagem real do desempenho. Para resolver isso, a empresa introduziu recursos inovadores, como pressões e quadros congelados para remates, bem como altura de impacto do remate, para melhorar a análise de ações

defensivas e criar modelos xG mais representativos. Essas ferramentas permitiram que os analistas entendessem e avaliassem melhor o desempenho, fornecendo-lhes os recursos necessários para realizar seus trabalhos com eficiência (StatsBomb, n.d.-a).

2.7.5 WYSCOUT

A Wyscout é uma empresa de tecnologia amplamente utilizada para prospeção, análise de partidas e avaliação de desempenho no futebol. Desde a sua criação em 2004, a Wyscout tem trabalhado para promover o desenvolvimento do futebol, apoiando clubes e jogadores individuais. A empresa oferece uma gama de produtos que reúnem vídeos, dados e informações sobre jogadores, competições e jogos (Football Analytics 101, 2019).

A base de dados de vídeos de futebol da Wyscout é a maior do mundo, com mais de 2.000 novos jogos carregados todas as semanas. Esses jogos são analisados pelos analistas da Wyscout e segmentados em mais de 2.000 videoclipes marcados e fáceis de encontrar. Com 400 milhões de jogadas cobrindo uma ampla gama de competições, incluindo os 5 principais campeonatos europeus e importantes torneios juvenis, a Wyscout analisa mais de 250 competições de futebol todas as semanas. Os utilizadores podem selecionar ações para equipas, jogadores ou jogos específicos e assistir a vídeos relacionados conforme sua conveniência. Eles também podem fazer *download* de clipes e criar a sua própria análise de vídeo, bem como listas de reprodução personalizadas que podem ser compartilhadas com outros profissionais do futebol (Wyscout, 2022).

A plataforma da Wyscout foi projetada tendo em mente as necessidades específicas de diferentes profissionais do futebol, incluindo agentes, olheiros, treinadores, árbitros, jogadores e jornalistas. A empresa oferece uma gama de produtos e pacotes adaptados às necessidades desses diferentes grupos, incluindo ferramentas para prospeção e identificação de talentos, análise tática, avaliação de desempenho e jornalismo baseado em dados. A plataforma da Wyscout é um recurso valioso para profissionais de futebol que buscam aprimorar suas habilidades e conhecimentos e manterem-se atualizados com os últimos desenvolvimentos do desporto (Wyscout, 2022).

2.7.6 Panoris

A CamVision, uma empresa tcheca de tecnologia fundada em 2007. É especializada em sistemas inteligentes de captura de vídeo para eventos desportivos. Com um forte foco em pesquisa e desenvolvimento no campo de vídeos desportivos, a Camvision conseguiu alavancar a tecnologia de ponta para fornecer soluções automatizadas para análise de vídeo. O seu produto, PANORIS, é capaz de registar e analisar cada movimento num evento desportivo de forma autônoma (Football Analytics 101, 2019).

PANORIS oferece uma variedade de recursos para gravar e analisar partidas de futebol. O sistema oferece cobertura totalmente panorâmica, "bandeira a bandeira" de todo o campo e pode ser configurado para gravar automaticamente ou de forma programada. O fluxo de vídeo em tempo real e vários tipos de visualização, incluindo gol a gol, box a box e amplas predefinições panorâmicas, permitem uma análise detalhada da jogada. O sistema também inclui funcionalidade de marcação de eventos, bem como rastreamento de jogadores e ferramentas gráficas para melhor orientação e instruções mais claras (CamVision, n.d.).

Além desses recursos, o produto da CamVision oferece a capacidade de exportar e compartilhar vídeos de alta resolução para uso em reuniões de equipe e sessões de estratégia. O sistema também é compatível com outras ferramentas de análise desportiva e permite o compartilhamento em plataformas de mídia social como YouTube, Facebook e Twitter. Também tem a capacidade de adicionar o placar, estatísticas ou conteúdo publicitário diretamente ao vídeo por meio de uma sobreposição de gráficos. O sistema também tem a capacidade de transmitir em tempo real para ferramentas de terceiros (CamVision, n.d.).

2.7.7 Champdas

Champdas é uma empresa desportiva, fundada em 2016 em Xangai na China, que possui e opera os seus próprios sistemas proprietários de coleta de dados, mineração de dados e produção de dados. Champdas é capaz de coletar mais de 15 mil pontos de dados por jogo em tempo real, que são apresentados na sua plataforma de dados, Champdas DATA. A empresa tem foco principal nos eventos domésticos, ou seja, transmissões ao vivo da Superliga e Liga Chinesa, e também acompanha de perto o desempenho da seleção nacional chinesa (Champdas, n.d.).

2.7.8 Tongdaoweiye

Estabelecida em 2015 em Pequim, China, a Tongdaoweiye é uma empresa de *big data* de futebol que fornece uma variedade de serviços, incluindo coleta de dados, análise de modelagem, aplicação de dados e gerenciamento de sistemas de informações. Oferecem, também, serviços de dados e serviços de transferência e afirmam ter desenvolvido uma variedade de tecnologias, incluindo uma plataforma de gerenciamento e plataforma de dados. Tongdaoweiye estabeleceu parcerias com a Associação Chinesa de Futebol e equipas de futebol profissionais chineses (Tongdaoweiye, n.d.).

2.7.9 SkillCorner

SkillCorner é uma startup de visão computacional, fundada em Paris, em 2016, que desenvolveu uma tecnologia de rastreamento de vídeo com IA baseada em DL. Esta tecnologia é capaz de reconhecer, posicionar e acompanhar os jogadores de futebol, o árbitro e a bola em tempo real durante as transmissões ao vivo. O algoritmo de rastreamento da SkillCorner tem uma taxa de precisão de até 95%, tornando-o um dos melhores atualmente. De dados brutos de rastreamento à visualização de jogos de futebol, os produtos da SkillCorner estão na vanguarda da inovação tecnológica (SkillCorner, n.d.).

2.7.10 Comparação das Soluções

Como está explícito na tabela 3, muitas funcionalidades são comuns a todas as soluções, tal como a análise da performance, através das métricas e representações gráficas necessárias. Porém apesar de algumas delas apresentarem a componente de recrutamento de atletas, nenhuma solução tem a funcionalidade de recrutamento de treinadores.

Tabela 4: Comparação Funcionalidades das Soluções Existentes

	SAP-Sports-One	InStat Scout	Opta	StatsBomb IQ	WYSCOUT	PANORIS	Champdas	Tongdaoweiye	SkillCorner
Gerenciamento Equipa	✔	✔		✔	✔			✔	✔
Gerenciamento Treino	✔							✔	
Acompanhamento Bem-Estar Atleta	✔	✔							
Recrutamento Atletas	✔	✔	✔	✔	✔				
Recrutamento Treinadores									
Análise Performance	✔	✔	✔	✔	✔		✔	✔	✔
Análise Vídeo	✔	✔	✔		✔	✔	✔	✔	✔
Desktop App	✔		✔	✔	✔	✔		✔	✔
Web App		✔	✔	✔	✔		✔	✔	
Mobile App	✔	✔	✔	✔	✔	✔	✔	✔	
Preço	Não divulgado	35€ - 130€ /mês	Não divulgado	Não divulgado	270€ - 678€ /ano	Não divulgado	Não divulgado	Não divulgado	Não divulgado
Período Trial	14 dias	Sim	Não	Sim	15 dias	Não	Não divulgado	Não divulgado	Não divulgado

Este projeto pretende inovar na questão de recrutamento de treinadores, bem como manter também as funcionalidades mais comum, análise de performance. O conjunto de requisitos funcionais podem ser encontrados no capítulo 3 deste documento.

2.8 Ética, Privacidade e Segurança

O processamento de dados relacionados ao futebol levanta uma série de questões éticas, de privacidade e segurança. À medida que mais e mais dados são coletados sobre jogadores, equipas e partidas de futebol, é importante considerar os possíveis impactos sobre os indivíduos e o desporto como um todo.

Uma preocupação ética fundamental é o uso de dados para fins comerciais. Clubes de futebol e outras organizações podem coletar e usar dados sobre jogadores e partidas para vender bilhetes, mercadorias e outros produtos relacionados. Isso levanta questões sobre o uso justo de dados e até que ponto os jogadores e outras partes interessadas são capazes de controlar como os seus dados são usados (União Europeia, 2016).

Outra preocupação é o potencial de os dados serem usados para obter uma vantagem que pode ser vista como injusta, na competição. Por exemplo, as equipas podem usar dados para identificar pontos fracos nos seus oponentes e desenvolver táticas para explorá-los. Isso pode ser considerado como desigual, para os mais conservadores, e prejudicar a integridade do desporto.

A privacidade é outra consideração importante quando se trata de dados relacionados ao futebol. Jogadores e outras partes interessadas podem se preocupar com a coleta e uso dos seus dados pessoais, incluindo informações confidenciais, como registos de lesões e histórico médico. É importante que os coletores de dados e utilizadores sejam transparentes sobre suas práticas de tratamento de dados e garantam que os dados pessoais sejam processados de acordo com as leis e regulamentos relevantes (RGPD) (União Europeia, 2016).

A segurança também é uma questão crítica quando se trata de dados relacionados ao futebol. Esses dados podem ser valiosos para uma ampla gama de partes, incluindo hackers e outros cibercriminosos. É importante que os coletores de dados e utilizadores tenham medidas de segurança robustas para proteção contra acesso não autorizado, uso indevido e perda de dados.

No geral, é importante que a indústria do futebol considere cuidadosamente as implicações éticas, de privacidade e segurança do processamento de dados. Isso inclui tomar medidas para garantir que os dados sejam coletados e usados de maneira justa e responsável, e que os dados pessoais sejam protegidos e mantidos em segurança. Ao fazer isso, o setor pode criar confiança entre jogadores, fãs e outras partes interessadas e garantir que os dados sejam usados em benefício do desporto e dos seus participantes.

2.9 Resumo

Neste capítulo é apresentado o estado de arte sobre os conceitos considerados relevantes, que dizem respeito ao Futebol e à inteligência Artificial.

Primeiramente são apresentados os tipos de dados utilizados no futebol, incluindo dados de fluxo de eventos (informações coletadas e registadas durante um jogo) e dados de rastreamento ótico (informações registadas sobre os movimentos dos jogadores em campo). São também mencionadas as empresas que fornecem esses dados, como Opta e Prozone, e os desafios que surgem da diversidade de *providers* de dados, como falta de informação, terminologia e definições diferentes e a dificuldade de aplicar ferramentas de análise automática.

De seguida, é apresentado o conceito da análise avançada no futebol, que envolve o uso de técnicas complexas de análise de dados, como aprendizagem automática e visualização de dados, para extrair informações valiosas e melhorar o desempenho. Os conceitos de posse e

sequência, que são centrais para análises avançadas, são descritos. A posse de bola é definida como uma ou mais sequências seguidas pertencentes à mesma equipa, enquanto uma sequência é uma passagem de jogo que pertence a uma equipa e é finalizada por ações defensivas, interrupções no jogo ou remates. Através da análise das sequências é possível perceber padrões e classificá-los de acordo com esses padrões. Os tipos de sequências explicados são a manutenção da posse de bola, construção do jogo, ameaça contínua, ritmo rápido, jogo direto, contra-ataque, cruzamento e pressão alta.

Após isto, o foco passa para os conceitos relativos à IA. São descritos os conceitos de aprendizagem automática (ML) e aprendizagem profunda (DL), que são subcampos da inteligência artificial que envolvem o projeto. E é também dado especial foco e são descritos os modelos mais utilizados nesta área da análise desportiva, como as RNA, Autoencoders e GANs. É explicado que existem vários tipos de ML, incluindo aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço. Quando se trata de futebol, a IA está a ser usada para melhorar muitos aspetos da indústria, como rastrear as posições dos jogadores durante as transmissões, automatizar o conteúdo do jogo e identificar futuras estrelas.

De seguida, são descritas as soluções existentes para análise de futebol, incluindo InStat Scout, Opta, StatsBomb IQ e Wyscout. Estas plataformas fornecem uma variedade de ferramentas e recursos, como visualização de dados, análise de desempenho, análise de vídeo e gerenciamento da equipa para ajudar treinadores, analistas e outras partes interessadas a obter informações sobre o desempenho da equipe e do atleta e tomar decisões informadas. São amplamente utilizados por vários clubes de futebol ao redor do mundo e são conhecidos pelas suas métricas avançadas e análises baseadas em dados.

Por fim, são discutidas as questões éticas, de privacidade e segurança que surgem com o processamento de dados relacionados ao futebol. São destacadas as preocupações sobre o uso de dados para fins comerciais e o potencial de serem usados para obter uma vantagem desleal na concorrência. São abordadas as preocupações sobre privacidade e proteção de dados pessoais, bem como a necessidade de segurança de dados para proteção contra acesso não autorizado e uso indevido.

3 Análise e Desenho da Solução

O processo de definição, documentação e manutenção dos requisitos é denominado Engenharia de Requisitos (ER) (Chemuturi, 2013). Durante a fase de Análise, os desenvolvedores e analistas coletam informações para a solução de um problema específico e para alcançar os objetivos predefinidos. Esse processo requer uma comunicação ativa entre o desenvolvedor e o cliente ou utilizadores do sistema, uma vez que o cliente detém mais conhecimento sobre o sistema necessário e os utilizadores serão aqueles que utilizarão o sistema. Reuniões frequentes entre esses elementos permitem a antecipação dos requisitos funcionais e não funcionais, reduzindo a lacuna de conhecimento entre as partes interessadas e garantindo que o produto final atenda às expectativas.

Os requisitos funcionais especificam as funcionalidades do sistema e definem todas as interações possíveis com os utilizadores. Por exemplo, "Um utilizador deve ser capaz de se registar" ou "Um utilizador deve ser capaz de alterar seu registo" são requisitos funcionais que representam interações entre o utilizador e o sistema.

Por outro lado, os requisitos não funcionais não estão necessariamente relacionados às funcionalidades ou regras de negócio do sistema na perspetiva do utilizador, mas correspondem a restrições que o sistema deve atender. Os desenvolvedores usam esses requisitos para tomar decisões de arquitetura, implementação e desempenho. Alguns exemplos incluem a seleção de versões de tecnologias devido a requisitos de compatibilidade ou a escolha de uma estrutura de *backend* diferente devido à sua rapidez.

Em resumo, essa secção aborda a análise do projeto e o processo de design, utilizando práticas adequadas.

3.1 Domínio do Problema

A figura 13 descreve uma representação de alto nível dos conceitos que englobam este projeto e da forma como estes se relacionam.

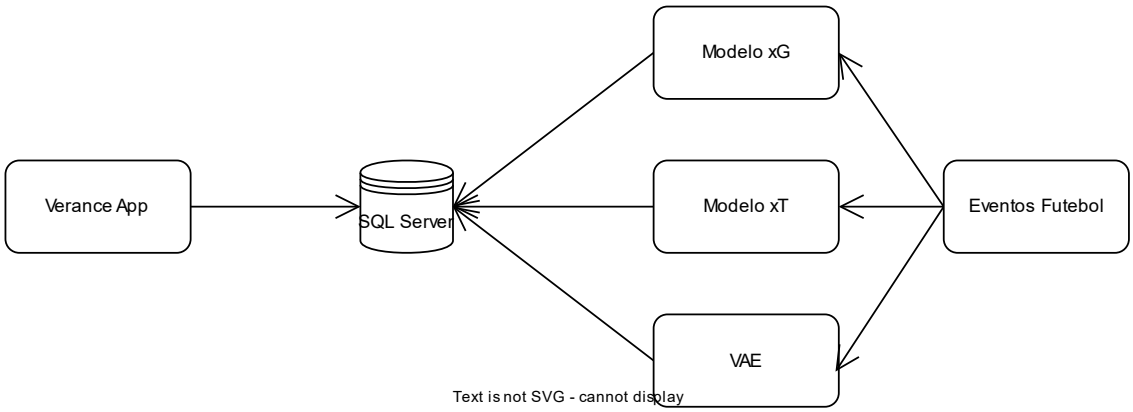


Figura 13: Arquitetura de Alto Nível da Verance App

Existe um processo automático, ou seja, um "*job*", que é executado semanalmente. Esse *job* tem como objetivo fornecer eventos de futebol para os diferentes modelos de IA, que integram este projeto.

Após receber os eventos de futebol, os modelos são executados usando pipelines específicas para cada um deles. Os resultados gerados por cada modelo são armazenados numa base de dados SQL Server. Isto significa que cada vez que o *job* é executado, novos dados são adicionados à base de dados.

A Verance App utiliza os dados presentes na base de dados SQL Server para alimentar as suas visualizações. Os utilizadores finais podem ver os resultados dos diferentes modelos, que foram alimentados com os eventos de futebol da semana em questão. As visualizações incluem gráficos, tabelas e outros tipos de representação visual dos dados.

3.2 Stakeholders

Os *Stakeholders* ou partes interessadas são consideradas entidades que podem ser afetadas, afetam ou têm interesse no sistema. Portanto, é essencial identificar quem são essas entidades, bem como podem influenciar o processo de desenvolvimento do sistema. Para esta solução foram identificados os seguintes *stakeholders*:

- **Sistema** – entidade encarregue de apresentar os resultados do projeto como os stats das sequências e das equipas, resultados de xG e xT;
- **Utilizador** – entidade que utiliza a aplicação. Visualiza os resultados dos modelos treinados no projeto.

3.3 Requisitos

O modelo FURPS é usado na Engenharia de *Software* para definir requisitos, e o "+" foi adicionado posteriormente para incluir categorias adicionais. Essa versão expandida, conhecida como FURPS+, fornece uma definição mais clara e concisa dos requisitos, facilitando o entendimento durante as fases de análise do projeto. A tabela abaixo explica o significado de cada letra do modelo, incluindo o sinal "+" (Eeles, 2004).

Tabela 5: Modelo FURPS+

CATEGORIA	DESCRIÇÃO
FUNCIONALIDADE	Funcionalidades que o sistema deve suportar. Requisitos Funcionais.
USABILIDADE	Avaliação da interface com o utilizador, tanto do ponto de vista estético como de informação e documentação de suporte.
FIABILIDADE (DO INGLÊS **R**ELIABILITY)	Integridade e Conformidade do Software. Refere-se à duração máxima de inatividade aceitável, à frequência, gravidade e possibilidade de recuperação, em caso de falha.
DESEMPENHO (DO INGLÊS **P**ERFORMANCE)	Categoriza a rapidez do sistema, tempo máximo de resposta e consumo de recursos.
SUPORTABILIDADE	Agrupa características como: testabilidade, manutenibilidade, compatibilidade e escalabilidade.
+	Adiciona Requisitos de Qualidade como: restrições de design e implementação ou requisitos de hardware e interface.

3.3.1 Requisitos Funcionais

O modelo FURPS+ inclui requisitos funcionais na letra “F” conforme discutido anteriormente (Eeles, 2004). Cada requisito é identificado por um código alfanumérico para fácil referência. A Tabela 5 fornece uma lista dos requisitos funcionais identificados.

Tabela 6: Requisitos Funcionais para o Projeto

Identificador	Descrição
RF-01	Deve ser possível selecionar uma competição ou liga de futebol
RF-02	Deve ser possível selecionar uma equipa referente à competição anteriormente selecionada
RF-03	Deve ser possível visualizar as métricas avançadas da equipa selecionada
RF-04	Deve ser possível selecionar um jogo referente à equipa e competição selecionadas anteriormente
RF-05	Deve ser possível selecionar e visualizar uma sequência do jogo anteriormente selecionado
RF-06	Deve ser possível visualizar as métricas avançadas da sequência selecionada
RF-07	Deve ser possível selecionar uma ação da sequência anteriormente selecionada
RF-08	Deve ser possível visualizar as métricas avançadas da ação selecionada

3.3.2 Requisitos Não Funcionais

Os requisitos que não se relacionam diretamente com as funcionalidades do sistema são classificados como requisitos não funcionais e podem ser categorizados nas letras restantes do acrônimo FURPS+. (Eeles, 2004) apresenta a categoria "URPS+" para esses requisitos. A seguir, serão descritos mais detalhadamente os requisitos que se enquadram nessa categoria.

3.3.2.1 Usabilidade

De acordo com (Nielsen, 1993), a usabilidade pode ser analisada por meio de cinco atributos principais: capacidade de aprendizagem, precisão, velocidade de desempenho, capacidade de memorização e satisfação subjetiva. Com base nesses atributos, foram identificados os seguintes requisitos:

- Capacidade de aprendizagem: a interface deve ser simples, fácil e intuitiva;
- Precisão dos resultados: o sistema deve fornecer resultados precisos;
- Rapidez na apresentação dos resultados: o sistema deve apresentar os resultados em poucos segundos;
- Desempenho responsivo: o sistema deve ser responsivo e reduzir a latência;

- Satisfação do utilizador: o sistema deve proporcionar uma experiência agradável e satisfatória ao utilizador.

3.3.2.2 Fiabilidade

A fiabilidade de um sistema está relacionada com a sua capacidade de prevenir falhas durante a sua vida útil e com a sua capacidade de se recuperar de falhas, caso ocorram (Todinov, 2016). Com base nisso, foram identificados os seguintes requisitos:

- O sistema deve estar permanentemente disponível, e incluir rotinas de controle para reduzir o tempo de indisponibilidade dos servidores;
- Se possível, o sistema deve ter servidores redundantes para maximizar a disponibilidade.

3.3.2.3 Desempenho

Conforme descrito em (Smith & Williams, 2003), o desempenho é um fator crítico que pode determinar o sucesso ou fracasso na qualidade do software. A análise destes autores gira em torno do tempo de resposta, tempo de inicialização e encerramento, capacidade de memória e utilização do CPU. Para este sistema, os seguintes requisitos de desempenho foram identificados:

- O sistema deve ser fluído e responder rapidamente.

3.3.2.4 Suportabilidade

Quanto ao atributo de suportabilidade, este está relacionado com características como testabilidade, manutenibilidade, compatibilidade e escalabilidade, conforme (Eeles, 2004). Para este atributo, os seguintes requisitos foram identificados:

- O sistema deve ser escalável para acomodar um número crescente de ligas e temporadas;

- Se possível, todos os requisitos devem ser atendidos usando testes unitários e end-to-end.

3.3.2.5 Requisitos Adicionais

Os requisitos adicionais correspondem ao "+" do modelo FURPS+ e incluem requisitos de qualidade, como restrições de design e implementação ou requisitos de hardware e interface. Especificamente para este sistema, foram identificados os seguintes requisitos adicionais:

Requisitos de Design

- Todos os módulos devem ser capazes de usar uma única camada de persistência.

Requisitos de Implementação

- Adoção de boas práticas de design, como o uso de *clean architecture* para o *backend.*

Requisitos de Hardware

- Utilizar o GPU disponível para treinar os modelos e fazer as inferências dos modelos.

3.4 Casos de Uso

Foram definidos casos de uso para atender aos requisitos descritos na secção anterior. O diagrama a seguir ilustra os casos de uso e suas descrições.

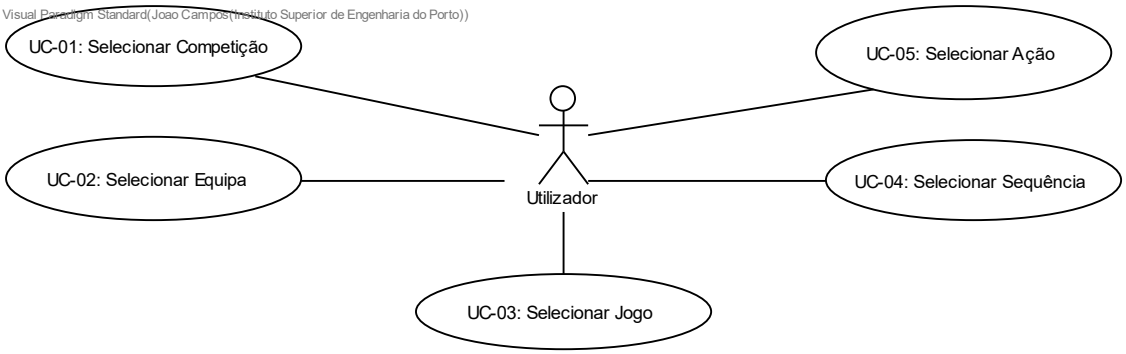


Figura 14: Diagrama de Casos de Uso

A seguir, serão apresentados todos os casos de uso, detalhadamente. Para isso, será utilizado um diagrama de sequência de sistema que ilustra visualmente o caso de uso. Em seguida, será realizada uma análise mais detalhada, que incluirá a identificação do ator, partes interessadas,

pré-condições, cenário principal, cenários alternativos possíveis, pós-condições e requisitos concretizados. Por fim, será incluída uma breve descrição do que foi pesquisado e estudado durante a análise de cada caso de uso, que servirá como introdução para a secção de design.

3.4.1 UC-01: Selecionar Competição

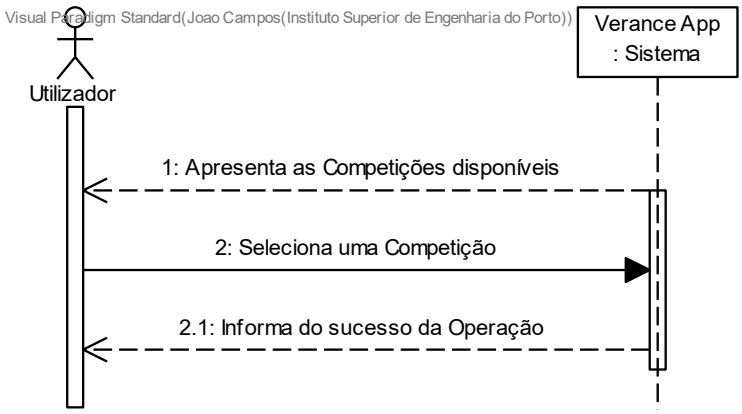


Figura 15: Diagrama de Sequência de Sistema para o UC-01

Ator Principal

O Ator deste caso de uso é o Utilizador.

Partes Interessadas e seus Interesses

Utilizador: Pretende selecionar a Competição com o intuito de selecionar as Equipas dessa mesma Competição.

Pré-Condições

É necessário que existam competições disponíveis na base de dados.

Cenário Principal

Como descrito na figura 15:

1. O sistema apresenta as competições disponíveis.

2. O utilizador seleciona a competição pretendida.
3. O sistema informa sobre o sucesso da Operação.

Cenários Alternativos

1a. Se ocorrer um erro na conexão com a base de dados, o sistema informa sobre uma falha na apresentação das competições disponíveis.

Pós-Condições

Depois de selecionada a competição, pode se aplicar o caso de uso UC-02, seleção da equipa referente à competição.

Requisitos Atendidos

Este caso de uso concretiza o requisito seguinte:

- RF-01: Deve ser possível selecionar uma competição ou liga de futebol.

Pesquisa e Estudo

Este caso de uso pode ser realizado mediante uma interface gráfica, em que a seleção da competição poderia ser mediante um *click* numa *dropdown.* Nesta *dropdown* estariam disponíveis todas as opções de temporadas, presentes na base de dados. Outra abordagem seria a listagem de todas as competições disponíveis numa tabela com o logotipo da competição e o nome da mesma e o utilizador escolheria a linha da tabela corresponde à competição pretendida.

3.4.2 UC-02: Selecionar Equipa

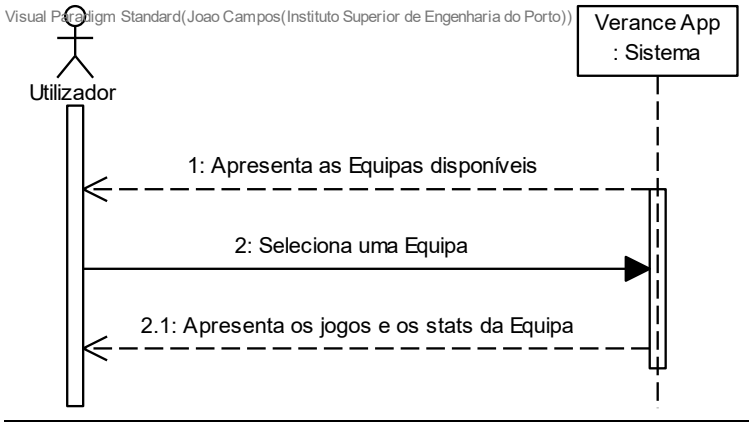


Figura 16: Diagrama de Sequência de Sistema para UC-02

Ator Principal

O Ator deste caso de uso é o Utilizador.

Partes Interessadas e seus Interesses

Utilizador: Pretende selecionar a Equipa com o intuito de selecionar os jogos dessa mesma Equipa.

Pré-Condições

É necessário que tenha sido selecionada uma competição previamente. É, também, indispensável que existam equipas registadas na base de dados.

Cenário Principal

Como descrito na figura 16:

1. O sistema apresenta as Equipas disponíveis, referentes à competição previamente selecionada.
2. O utilizador seleciona a equipa pretendida.

3. O sistema apresenta as estatísticas da equipa selecionada bem como os jogos da equipa, na competição, na temporada.

Cenários Alternativos

1a. Se ocorrer um erro na conexão com a base de dados, o sistema informa sobre uma falha na apresentação das equipas disponíveis.

Pós-Condições

Depois de selecionada a equipa, pode se aplicar o caso de uso UC-03, seleção do jogo referente à equipa, referente à competição.

Requisitos Atendidos

Este caso de uso concretiza o requisito seguinte:

- RF-02: Deve ser possível selecionar uma equipa referente à competição anteriormente selecionada.
- RF-03: Deve ser possível visualizar as métricas avançadas da equipa selecionada.

Pesquisa e Estudo

Em relação a este caso de uso, as equipas poderiam ser apresentadas numa tabela paginada e a sua seleção seria através de um *click* na linha correspondente à equipa desejada. Depois disso, o utilizador seria redirecionado para a página de detalhes da equipa, em que estariam presentes as estatísticas avançadas da equipa e os jogos da equipa selecionada, na competição previamente selecionada.

3.4.3 UC-03: Selecionar Jogo

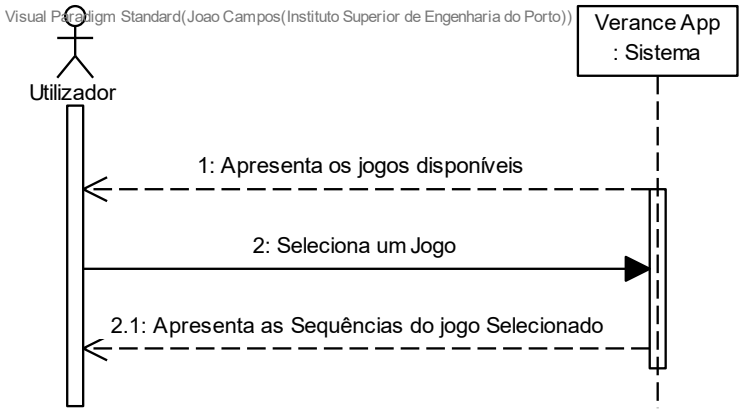


Figura 17: Diagrama de Sequência de Sistema para UC-03

Ator Principal

O Ator deste caso de uso é o Utilizador.

Partes Interessadas e seus Interesses

Utilizador: Pretende selecionar o jogo com o intuito de selecionar as sequências desse mesmo jogo.

Pré-Condições

É necessário que tenha sido selecionada uma equipa previamente. É, também, indispensável que existam jogos registados na base de dados.

Cenário Principal

Como descrito na figura 17:

1. O sistema apresenta os jogos disponíveis, referentes à equipa e competição previamente selecionados.
2. O utilizador seleciona o jogo pretendido.
3. O sistema apresenta as sequências referentes ao jogo selecionado.

Cenários Alternativos

1a. Se ocorrer um erro na conexão com a base de dados, o sistema informa sobre uma falha na apresentação dos jogos disponíveis.

Pós-Condições

Depois de selecionado o jogo, pode se aplicar o caso de uso UC-04, seleção das sequências referentes ao jogo.

Requisitos Atendidos

Este caso de uso concretiza o requisito seguinte:

- RF-04: Deve ser possível selecionar um jogo referente à equipa, temporada e competição selecionadas anteriormente.

Pesquisa e Estudo

No que concerne este caso de uso, os jogos poderiam ser apresentados numa tabela paginada e a sua seleção seria através de um *click* na linha correspondente ao jogo desejado. Depois disso, o utilizador seria redirecionado para a página de detalhes do jogo, em que estariam presentes as sequências do jogo selecionado.

3.4.4 UC-04: Selecionar Sequência

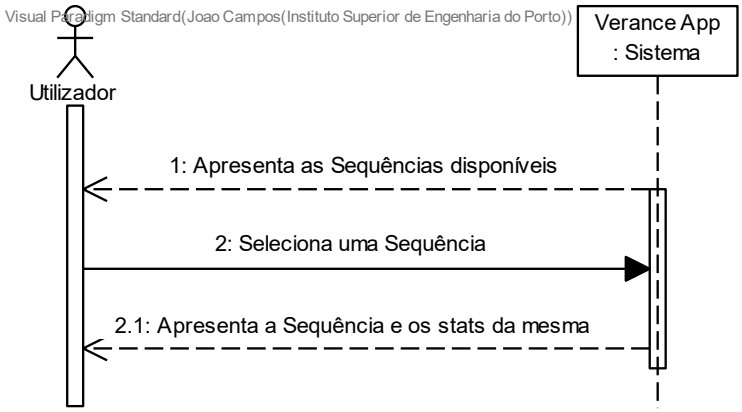


Figura 18: Diagrama de Sequência de Sistema para UC-04

Ator Principal

O Ator deste caso de uso é o Utilizador.

Partes Interessadas e seus Interesses

Utilizador: Pretende selecionar a sequência com o intuito de a visualizar e às suas estatísticas.

Pré-Condições

É necessário que tenha sido selecionado um jogo previamente. É, também, indispensável que existam sequências registadas na base de dados.

Cenário Principal

Como descrito na figura 18:

1. O sistema apresenta as sequências disponíveis, referentes ao jogo previamente selecionado.
2. O utilizador seleciona a sequência pretendida.
3. O sistema apresenta a sequência e as suas estatísticas.

Cenários Alternativos

1a. Se ocorrer um erro na conexão com a base de dados, o sistema informa sobre uma falha na apresentação das sequências disponíveis.

Pós-Condições

Depois de selecionada a sequência, pode se aplicar o caso de uso UC-05, seleção das ações referentes à sequência.

Requisitos Atendidos

Este caso de uso concretiza os requisitos seguintes:

- RF-05: Deve ser possível selecionar e visualizar uma sequência do jogo anteriormente selecionado.
- RF-06: Deve ser possível visualizar as métricas avançadas da sequência selecionada.

Pesquisa e Estudo

No que diz respeito a este caso de uso, as sequências poderiam ser apresentadas numa tabela paginada e a sua seleção seria através de um *click* na linha correspondente à sequência desejado. Depois disso, o utilizador seria redirecionado para a página de detalhes da sequência, em que seria apresentada uma visualização da respetiva sequência, bem como seriam apresentados os seus *stats*, entre eles, o xG e xT.

3.4.5 UC-05: Selecionar Ação

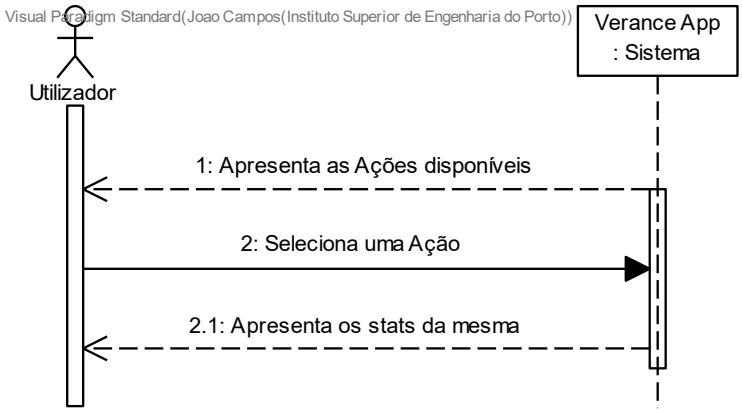


Figura 19: Diagrama de Sequência de Sistema para UC-05

Ator Principal

O Ator deste caso de uso é o Utilizador.

Partes Interessadas e seus Interesses

Utilizador: Pretende selecionar a ação com o intuito de visualizar as suas estatísticas.

Pré-Condições

É necessário que tenha sido selecionada uma sequência previamente. É, também, indispensável que existam ações registadas na base de dados.

Cenário Principal

Como descrito na figura 19:

1. O sistema apresenta as ações disponíveis, referentes à sequência previamente selecionada.
2. O utilizador seleciona a ação pretendida.
3. O sistema apresenta a ação e as suas estatísticas.

Cenários Alternativos

1a. Se ocorrer um erro na conexão com a base de dados, o sistema informa sobre uma falha na apresentação das ações disponíveis.

Pós-Condições

Não se aplicam.

Requisitos Atendidos

Este caso de uso concretiza os seguintes requisitos:

- RF-07: Deve ser possível selecionar uma ação da sequência anteriormente selecionada.
- RF-08: Deve ser possível visualizar as métricas avançadas da ação selecionada.

Pesquisa e Estudo

Este caso de Uso, semelhante ao anterior, as ações poderiam ser apresentadas numa tabela paginada e a sua seleção seria através de um *click* na linha correspondente à ação desejada. Depois disso, o utilizador seria redirecionado para a página de detalhes da ação, em que estariam apresentados os seus *stats*, entre eles, o xG e xT.

3.5 Desenho

A etapa de conceção da solução, que também é conhecida como design ou desenho, requer uma compreensão aprofundada do projeto, dos requisitos e dos objetivos. Na sequência, será apresentado o processo de design com base numa combinação entre o modelo "4+1" proposto por Philippe Krutchen em 1995 (Kruchten, 1995) e o modelo C4 proposto por Simon Brown (Brown, 2018).

3.5.1 Modelo “4+1”

Durante a etapa de desenho da solução, é comum ocorrerem ambiguidades devido às diferentes notações que podem ser utilizadas para representar um mesmo conceito. Por

exemplo, caixas podem ser usadas para representar programas, computadores físicos, casos de uso ou funcionalidades, e as anotações de fluxo, como setas e linhas, podem ser igualmente confusas. Além disso, muitas vezes certos aspetos são enfatizados em excesso em detrimento de outros. O modelo "4+1" procura solucionar essas questões por meio da separação em cinco visões complementares, como ilustrado na figura 20.

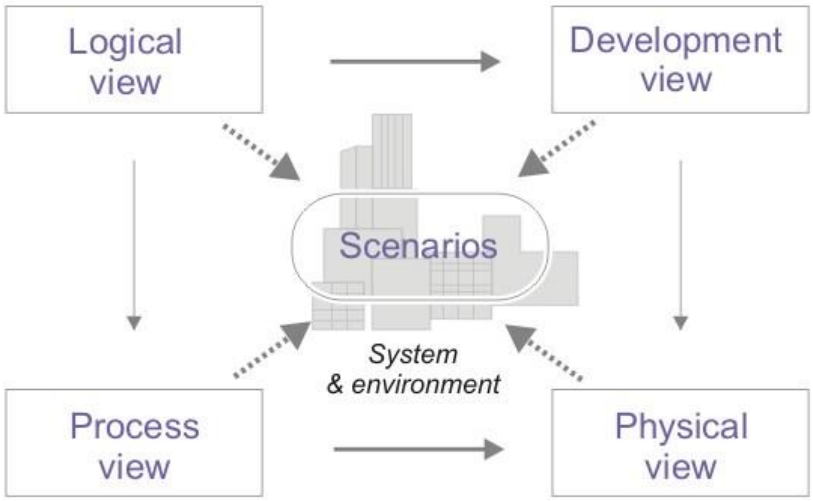


Figura 20: Arquitetura do Modelo "4+1" (Dekker, 2008)

A vista lógica é responsável por identificar e descrever os requisitos funcionais do sistema, ou seja, as suas funcionalidades e comportamentos. Essa visão procura entender como as regras de negócio devem ser atendidas e como o sistema se deve comportar em relação a elas. Os diagramas de componentes são apropriados para modelar essa visão, pois permitem visualizar as relações entre os componentes do sistema e como eles se interconectam (Kruchten, 1995).

A vista de processos é responsável por modelar o fluxo de processos ou interações do sistema, ou seja, como as informações e dados fluem dentro do sistema. Essa visão procura garantir a integridade do sistema e a sua capacidade de tolerância a falhas. Os diagramas de sequência são utilizados para modelar processos simples, enquanto os diagramas de estado são mais adequados para modelar fluxos mais complexos (Kruchten, 1995).

A visão de implementação, ou desenvolvimento, descreve a organização do software no ambiente de desenvolvimento, ou seja, como o software é dividido em pacotes, bibliotecas e módulos para facilitar o desenvolvimento, a manutenção e a evolução do sistema. Esta visão pode ser representada através de diagramas de pacotes, que mostram como as diferentes partes do sistema estão organizadas em camadas lógicas (Kruchten, 1995).

A vista física é responsável por descrever como os componentes do software são mapeados em hardware, ou seja, como o sistema é implementado em termos de hardware e onde ele é executado. Esta visão procura garantir que o sistema seja implementado numa arquitetura adequada, que possua os recursos necessários para a sua execução. Os diagramas de implantação são utilizados para modelar esta vista, mostrando como o software é distribuído em diferentes máquinas e como estas se conectam (Kruchten, 1995).

Por fim, a vista dos cenários descreve todos os casos de uso do sistema, ou seja, as diferentes maneiras pelas quais os utilizadores irão interagir com o sistema. Essa visão permite visualizar as funcionalidades que o sistema deve ter para atender às necessidades dos utilizadores. O diagrama de casos de uso é utilizado para modelar essa visão, mostrando como as diferentes funcionalidades do sistema estão relacionadas entre si. Esta vista já foi modelada na secção anterior (ver Figura 14).

3.5.2 Modelo C4

O Modelo C4, proposto por Brown (Brown, 2018), preconiza a descrição do software em quatro níveis de abstração: sistema, contentor, componente e código. Cada nível oferece maior detalhe sobre uma parte menor do sistema, de forma análoga a mapas geográficos. A vista de sistema corresponde ao globo, a vista de contentor corresponde a um mapa de cada continente, a vista de componentes corresponde a um mapa de cada país e a vista de código corresponde a um mapa de ruas e bairros de cada cidade. Esses níveis permitem contar diferentes histórias para diferentes audiências.

Os níveis são definidos da seguinte forma: nível 1 descreve o sistema como um todo, nível 2 descreve os contentores do sistema, nível 3 descreve os componentes dos contentores e nível 4 descreve o código ou partes menores dos componentes. O Modelo de Vista 4+1, por outro lado, apresenta o sistema por diferentes perspetivas, tais como lógica, de processos, de implementação, física e de cenários (Brown, 2018). Combinando esses dois modelos, é possível representar o sistema de várias perspetivas, cada uma com diferentes níveis de detalhe. Aqui vão ser apresentadas as vistas até ao terceiro nível de detalhe e vão ser ocultados alguns níveis considerados desnecessários.

3.5.3 Vista Lógica

3.5.3.1 Nível 1

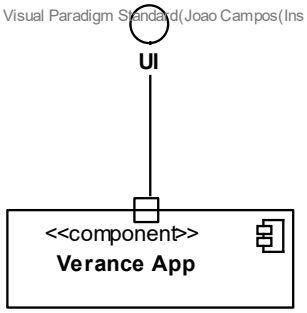


Figura 21: Diagrama de Componentes - Vista Lógica - Nível 1

3.5.3.2 Nível 2

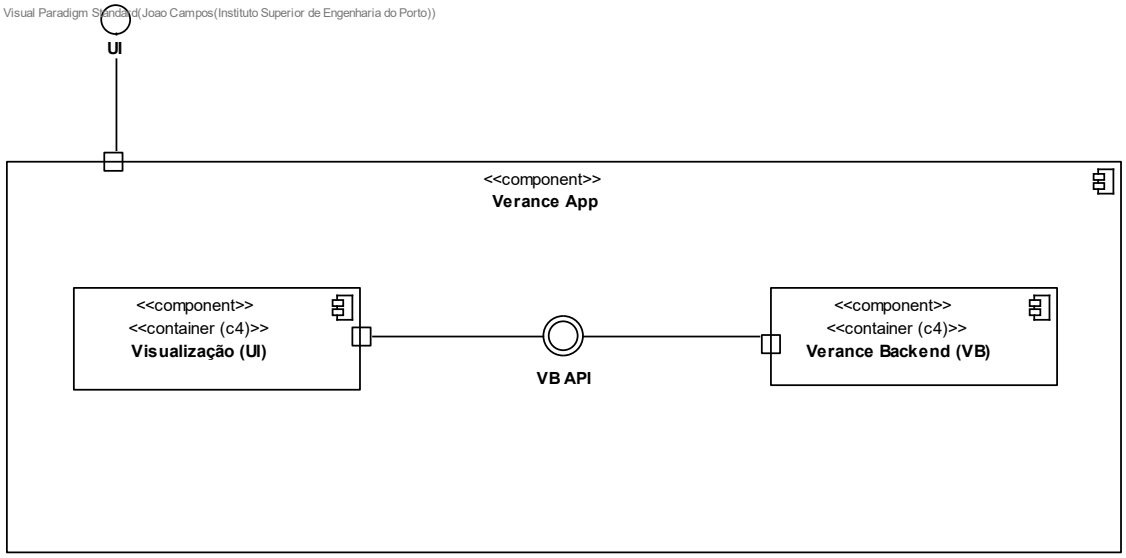


Figura 22: Diagrama de Componentes - Vista Lógica - Nível 2

3.5.3.3 Nível 3 (VB)

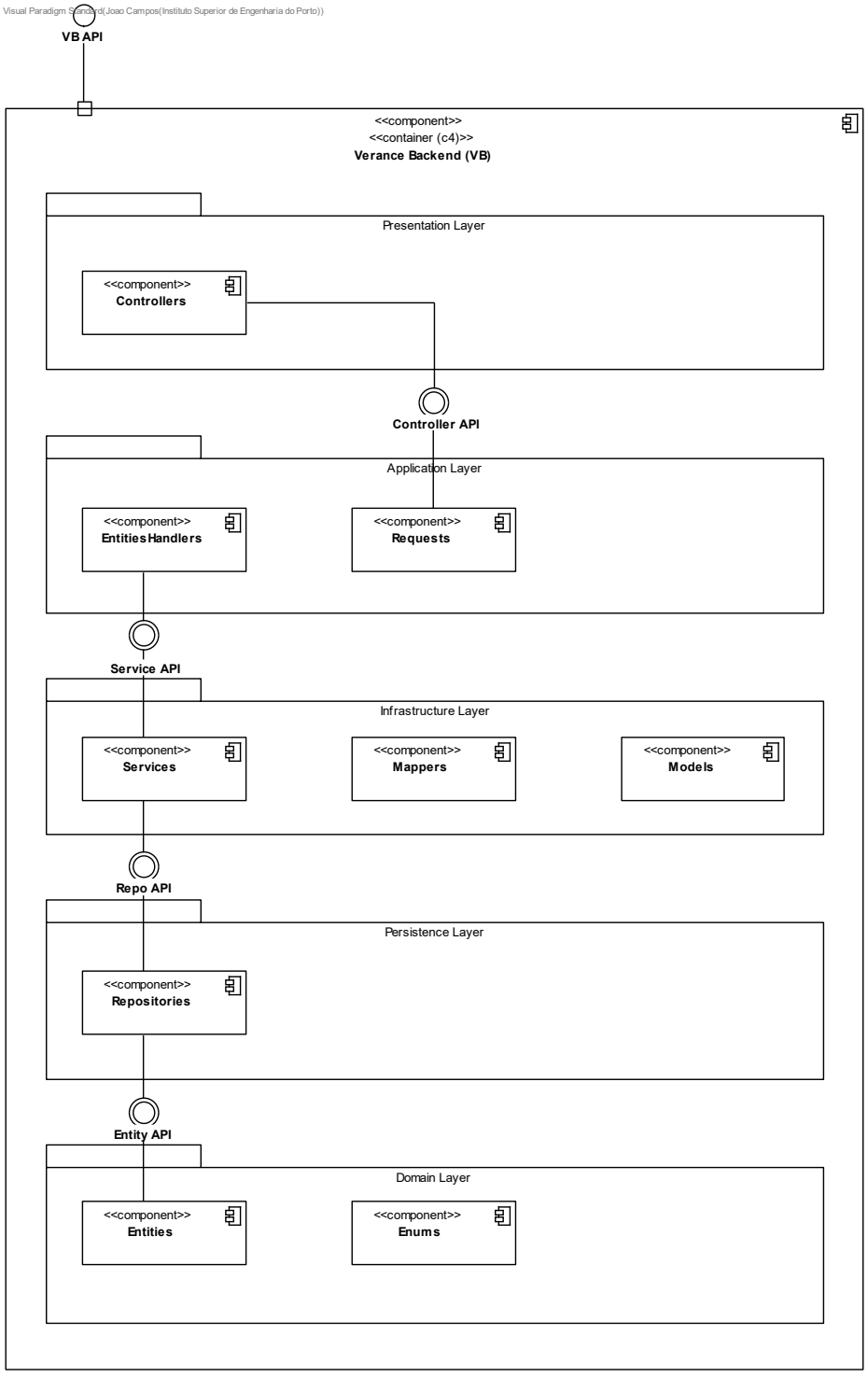


Figura 23: Diagrama de Componentes - Vista Lógica - Nível 3 de Verance Backend

3.5.3.4 Nível 3 (UI)

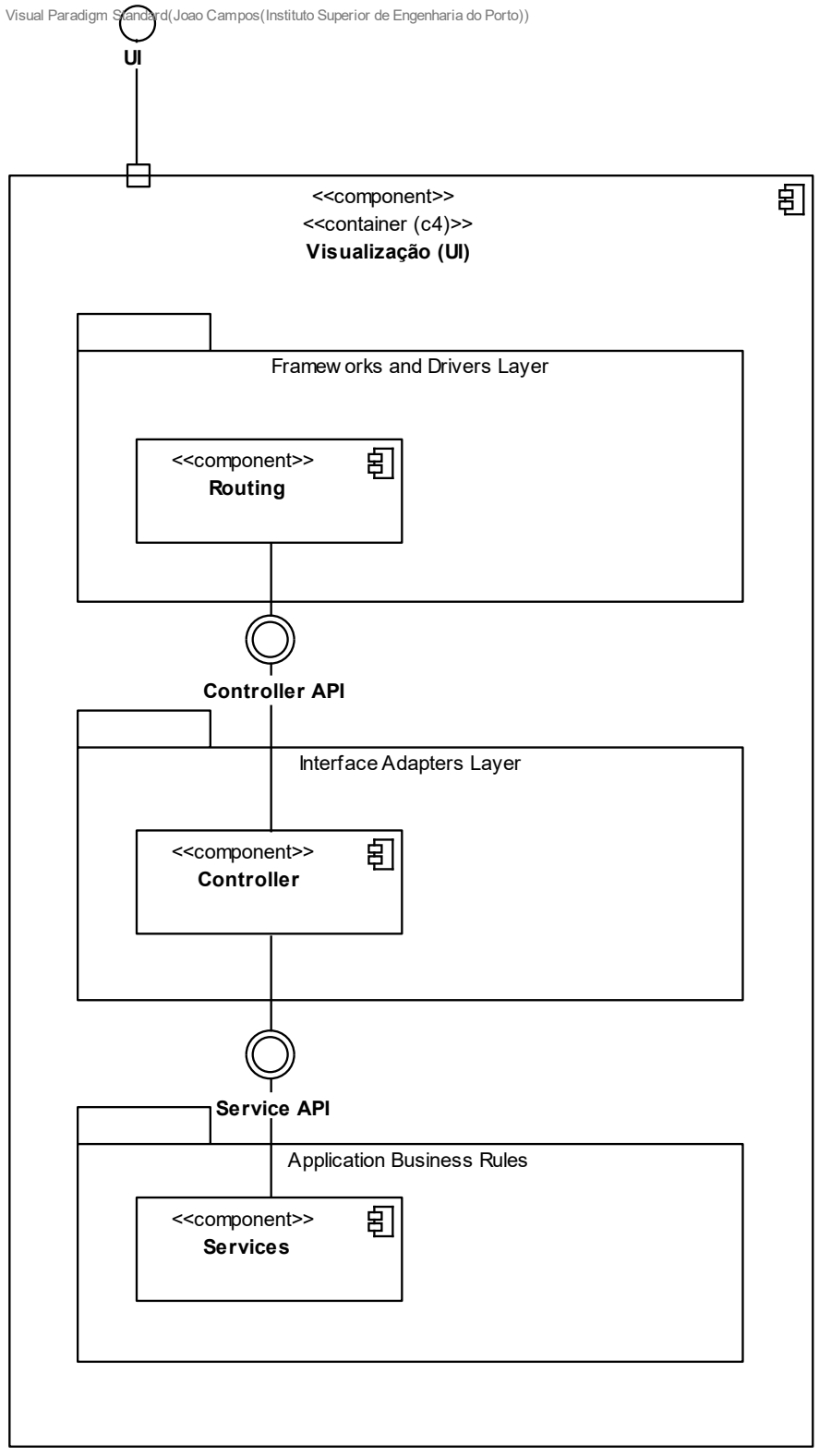


Figura 24: Diagrama de Componentes - Vista Lógica - Nível 3 de UI

Nos diagramas acima é possível perceber que, a cada nível de granularidade, é apresentada uma visão mais detalhada da solução. O Utilizador interage com a UI Utilizador e é nela que este seleciona as opções que pretende. O componente de Visualização UI, que é constituído pelas camadas de Drivers, Interface e de Regras de Negócio, comunica com o Verance Backend que segue uma Arquitetura de Camadas que encapsula a lógica de Negócio, conhecida como *Clean Architecture*. Nesta arquitetura separou-se a solução em 5 camadas:

- Na camada mais superficial temos a camada de Apresentação, constituída pelos Controllers que fazem uso de uma *design pattern* chamada *Mediator*. *Mediator* é um padrão de projeto comportamental que visa reduzir o acoplamento entre objetos ao limitar as suas interações diretas. Em vez disso, o padrão propõe a criação de um objeto mediador que gerência e coordena as interações entre esses objetos. O mediador atua como um intermediário entre os objetos. Recebe notificações de alterações e coordena as ações necessárias para manter a consistência do sistema. Isto permite que os objetos sejam desacoplados uns dos outros, já que não precisam conhecer a implementação de outros objetos com os quais interagem diretamente;
- A segunda camada aqui chamada de camada de Aplicação, possui os *Requests* e os *Handlers* de cada Pedido conhecido. A camada anterior cria um objeto de *Request* que vai acionar o *handler* da presente camada que depois interconecta com os serviços da camada seguinte;
- A camada de Infrastrutura possui todos os serviços, *mappers* que fazem a conexão entre as entidades e os modelos, e é a que acede aos repositórios;
- A penúltima camada, a camada de Persistência, é constituída pelos Repositórios das Entidades;
- A camada mais profunda possui todas as entidades que fazem parte da solução.

3.5.4 Vista de Processos

3.5.4.1 Nível 1

Os diagramas de Sequência deste nível de granularidade estão presentes no subcapítulo anterior, mais precisamente o subcapítulo 3.4.

3.5.4.2 Nível 2

A seguir são apresentados os diagramas de sequência referentes aos casos de uso identificados.

UC-01: Selecionar Competição

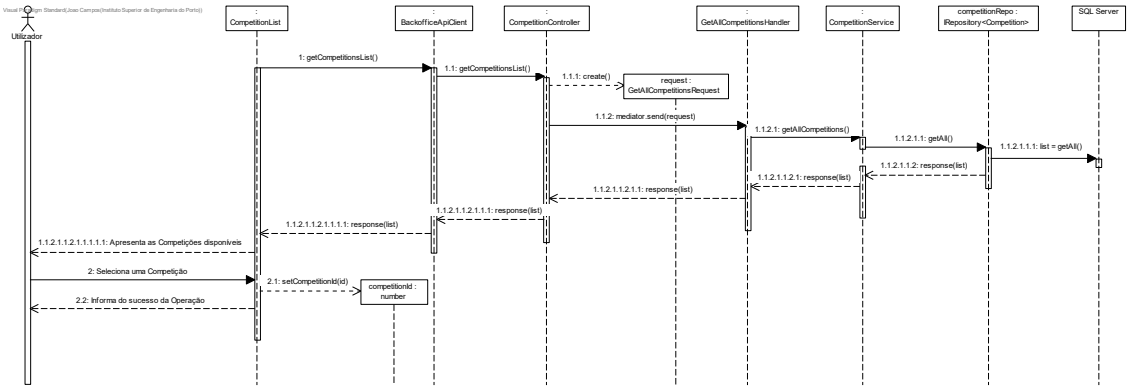


Figura 25: Diagrama de Sequência para UC01

Relativamente ao UC-01, na figura 25, é possível observar o processo de carregamento das Competições na página de listagem quando o utilizador acessa essa funcionalidade. O *frontend* envia uma solicitação ao *backend* requisitando todas as competições disponíveis. Em seguida, o CompetitionController cria e envia um pedido de listagem para o serviço. O *handler* é responsável por reencaminhar o pedido para o serviço que, por sua vez, acessa o repositório de dados para retornar todas as competições disponíveis. Após a conclusão dessa etapa, as mesmas são retornadas ao *frontend*.

Com as competições disponíveis, o *frontend* apresenta-as para o utilizador. Em seguida, o utilizador pode selecionar uma competição de interesse, que é armazenada num estado (*state*) para referência futura. O utilizador é então notificado sobre o sucesso da operação de seleção da competição.

UC-02: Selecionar Equipa

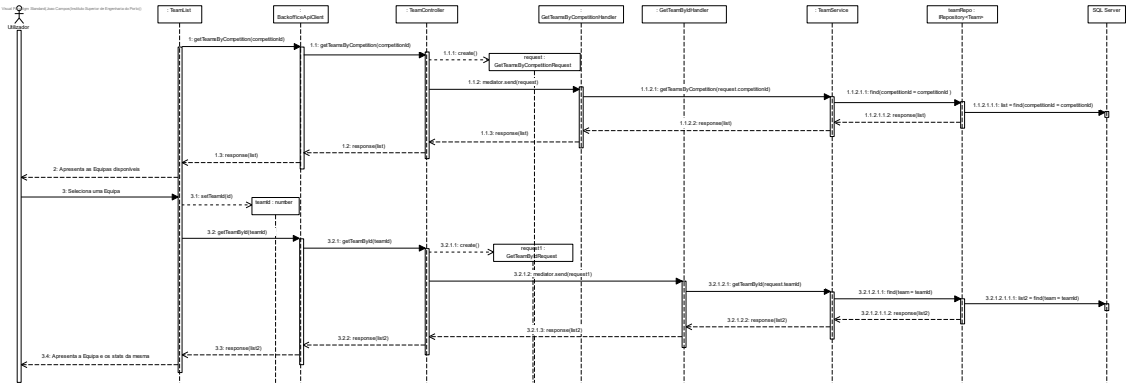


Figura 26: Diagrama de Sequência para UC02

A figura 26 ilustra o processo de carregamento das equipas na página de listagem quando o utilizador acessa essa funcionalidade, conforme descrito no caso de uso UC-02. Quando o utilizador entra na página de listagem das Equipas, o *frontend* envia uma solicitação ao *backend* requisitando todas as equipas disponíveis, para a competição selecionada, cuja referência estava guardada num *state*. O TeamController é responsável por criar e enviar um pedido de listagem para o *handler*, que, por sua vez, é encaminhado para o serviço. O TeamService então, acessa o repositório de dados para procurar todas as equipas disponíveis, para a competição selecionada, e as mesmas são retornadas ao *frontend*.

Com as equipas disponíveis, o *frontend* apresenta-as para o utilizador, que pode selecionar uma equipa de interesse. Similarmente aos casos de uso anteriores, a equipa selecionada é armazenada num estado (*state*). Após a conclusão da seleção da Equipa, o id da equipa selecionada é utilizado para retornar os detalhes da mesma. Nestes detalhes estão presentes as estatísticas da mesma.

UC-03: Selecionar Jogo

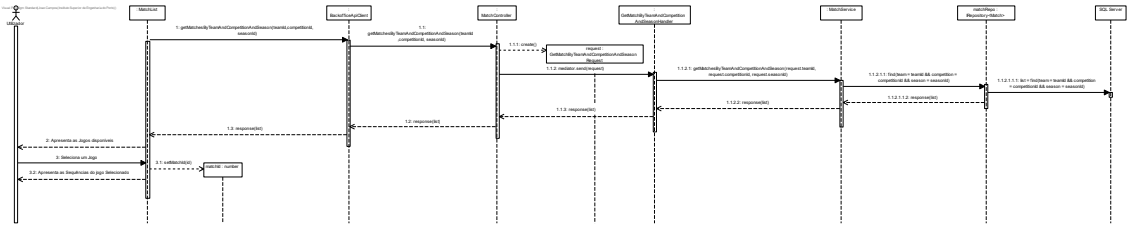


Figura 27: Diagrama de Sequência para UC03

No que toca ao caso de uso UC-03, a figura 27 apresenta o processo detalhado de carregamento dos jogos na página de listagem. Quando o utilizador entra na página, o *frontend* envia uma solicitação ao *backend* requisitando todos os jogos disponíveis para a equipa e a competição selecionadas, que são obtidas a partir de referências guardadas num estado (*state*). Como nos casos de uso anteriores, o *controller* da entidade em causa, neste caso o MatchController é responsável por criar e enviar um pedido de listagem para o *handler*, que encaminha para o serviço. O MatchService, por sua vez, acessa o repositório de dados para procurar todos os jogos disponíveis para a equipa e a competição selecionadas, e os mesmos são retornados ao *frontend*.

Com os jogos disponíveis, o *frontend* apresenta uma lista para o utilizador, que pode selecionar um jogo que o mesmo tenha interesse. Assim como nos casos de uso anteriores, o jogo selecionado é armazenado num estado (*state*) para ser usado no UC-04. Após isto o utilizador é notificado sobre o sucesso da operação.

UC-04: Selecionar Sequência

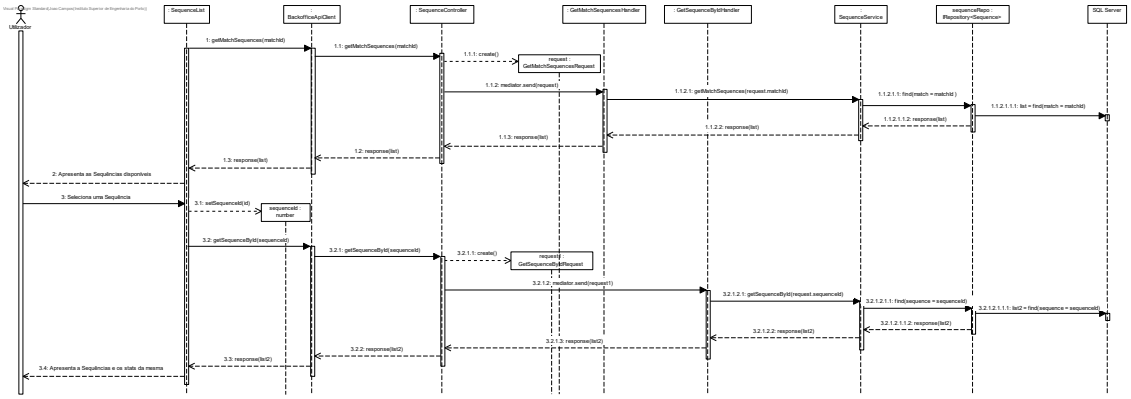


Figura 28: Diagrama de Sequência para UC04

Como demonstrado pela figura 28, quando o utilizador entra na página de detalhes do jogo, as sequências são carregadas. É utilizado o jogo selecionado para retornar as sequências do mesmo. Esta lógica vai desde o *frontend*, *controller*, *handler* e passa também pelo serviço e repositório. As sequências são retornadas até ao *frontend*.

De seguida, o utilizador, por sua vez, seleciona a sequência e o id desta é enviado para o *backend*, com o objetivo de serem apresentados os detalhes da mesma. No SequenceController é criado o *Request*, que é passado ao *Handler* e seguidamente ao Serviço. O SequenceService

vai ao Repositório procurar todas as ações da sequência pretendida, com os devidos detalhes. Todas estas ações são retornadas até ao *frontend* e estes detalhes são então apresentados ao utilizador pelo meio de gráficos e diagramas.

UC-05: Selecionar Ação

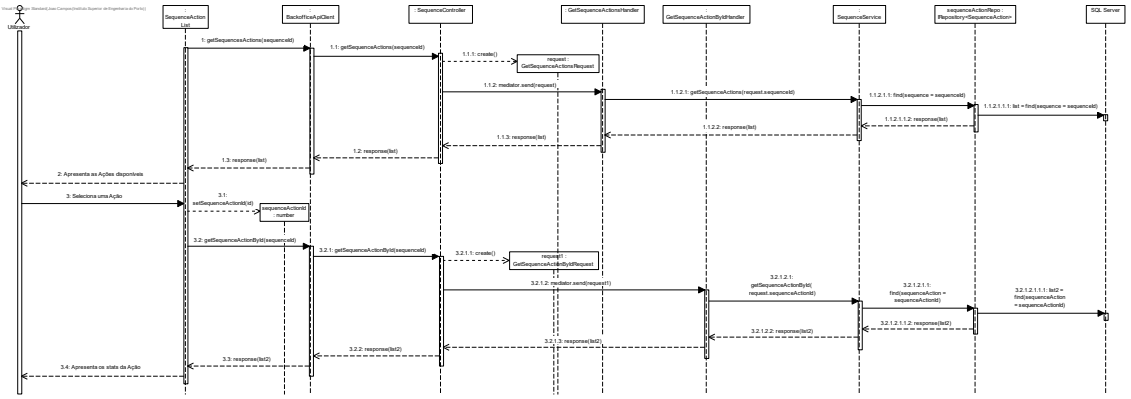


Figura 29: Diagrama de Sequência para UC05

Por fim, o último caso de uso UC-05 inicia-se com lógica semelhante a todos os outros. Neste caso é utilizada a sequência previamente selecionada para retornar as ações da mesma. Após o utilizador ver todas as ações, este pode selecionar uma delas. O id da mesma é então passado para que os detalhes desta sejam retornados. Todas estas ações são retornadas até ao *frontend* e estes detalhes são então apresentados ao utilizador por meio de visualizações.

3.5.5 Vista de Implementação

3.5.5.1 Nível 2

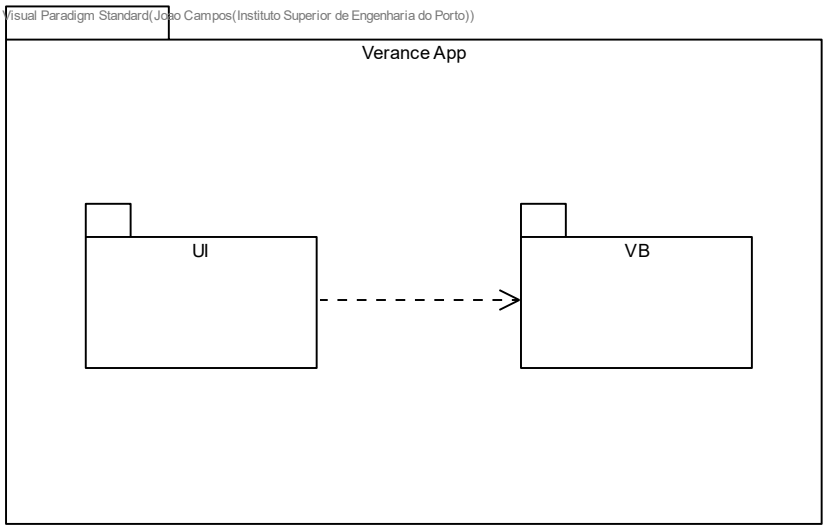


Figura 30: Diagrama de Pacotes - Vista de Implementação - Nível 2

3.5.5.2 Nível 3 (Verance Backend)

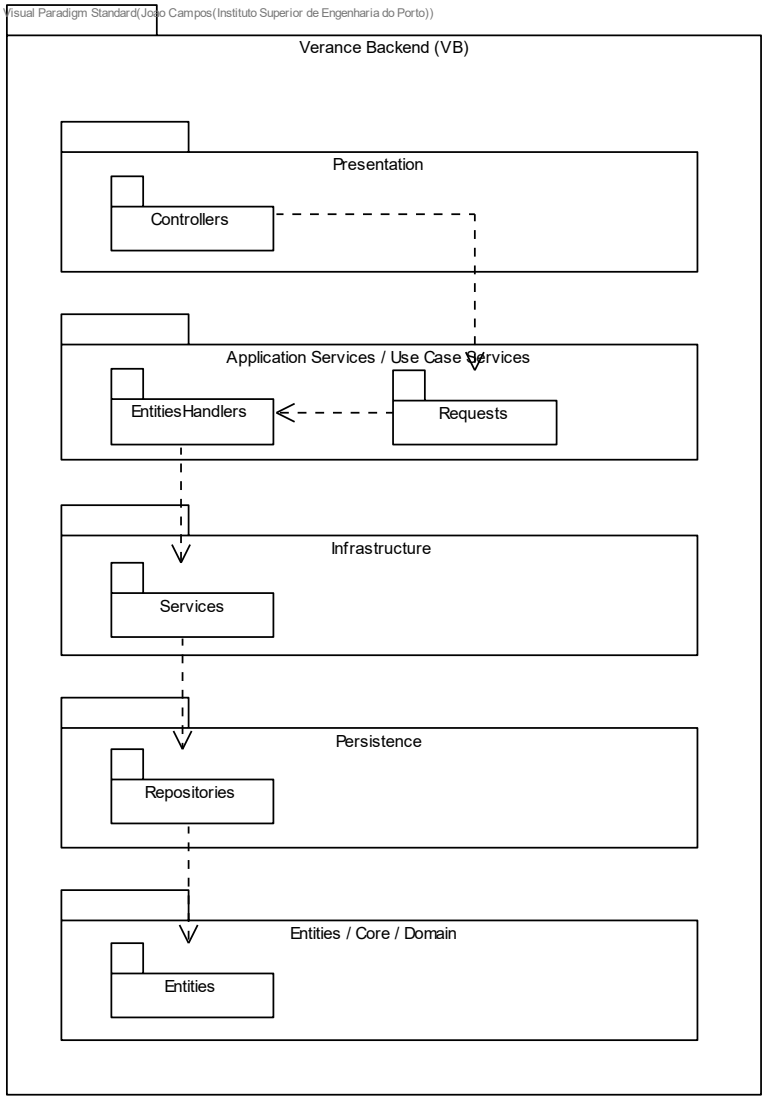


Figura 31: Diagrama de Pacotes - Vista de Implementação - Nível 3 de Verance Backend

3.5.5.3 Nível 3 (UI)

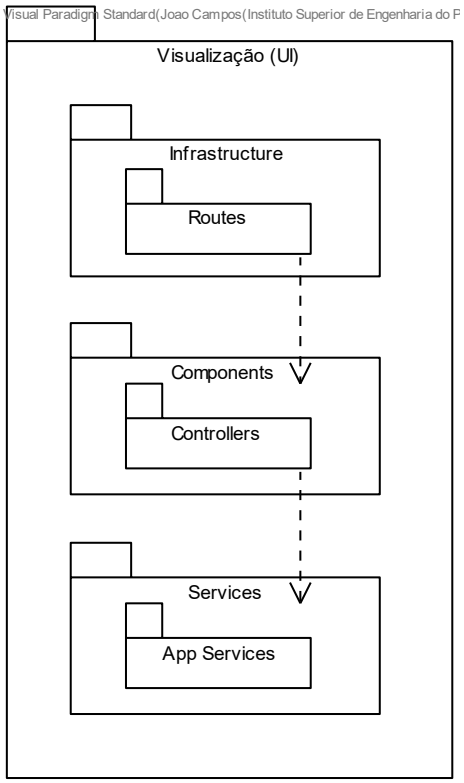


Figura 32: Diagrama de Pacotes - Vista de Implementação - Nível 3 de UI

Os diagramas acima ilustram que, à medida que se avança em cada nível de granularidade, uma visão mais detalhada da solução é apresentada. Os diagramas de Pacotes ilustram as dependências entre todos os componentes descritos anteriormente. As linhas representam relações diretas que identificam um pacote cujos membros devem ser importados.

3.5.6 Vista Física

3.5.6.1 Nível 2

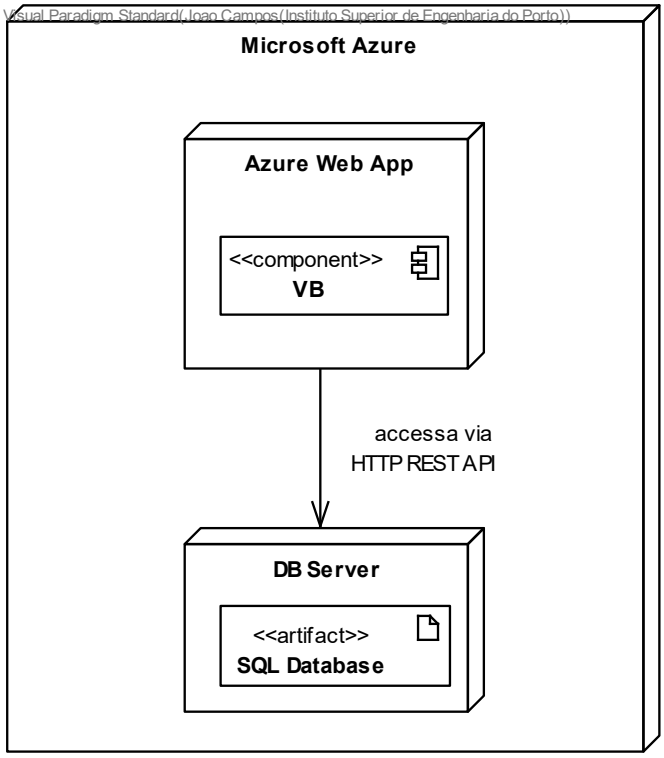


Figura 33: Diagrama de Implantação - Vista Física - Nível 2

3.5.6.2 Nível 3 (Verance Backend)

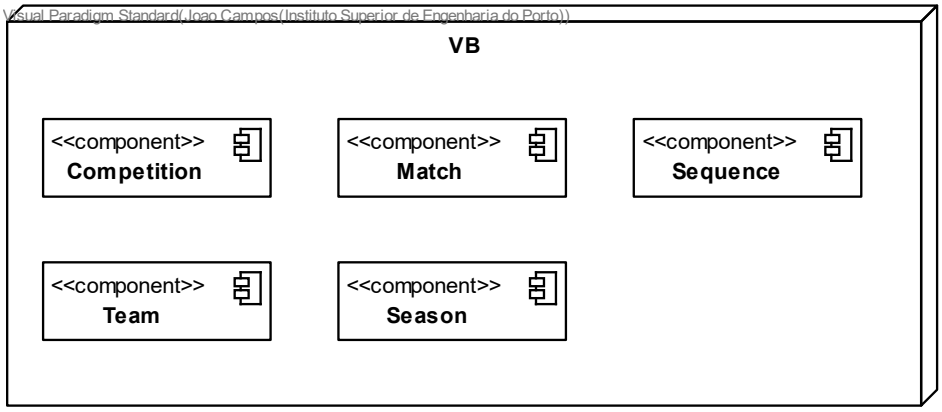


Figura 34: Diagrama de Implantação - Vista Física - Nível 3 de Verance Backend

3.5.6.3 Nível 3 (UI)

Sem necessidade de ser representado.

Os diagramas de Implantação anteriores modelam uma possível vista física deste projeto. O sistema está implantado como uma Azure Web App e alimenta-se dos dados de um servidor em SQL Server, também implantado no Microsoft Azure.

3.6 Desenhos Alternativos

A secção que se segue concentra-se numa possível alteração de design que poderiam ter levado a diferentes arquiteturas de software.

3.6.1 Vista Física Alternativa

No que concerne a esta mudança, seria no especto da possibilidade de implantação da solução.

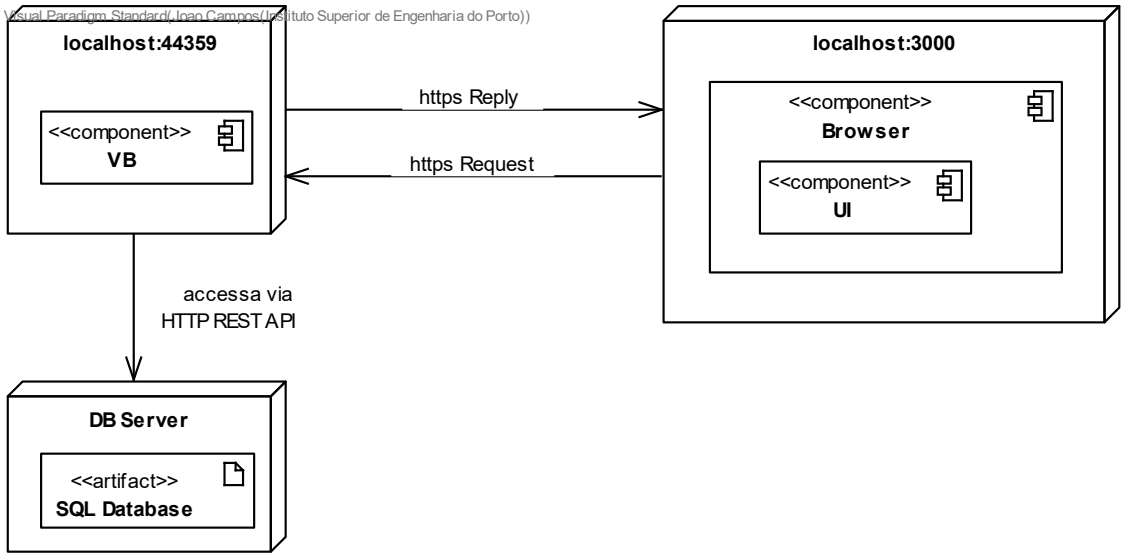


Figura 35: Diagrama de Implantação - Vista Física Alternativa

Nesta possibilidade o sistema era implantado localmente, contrariamente ao desenho alternativo, apresentado anteriormente. A UI faz pedidos ao *backend* que por sua vez, alimenta-se dos dados presentes num servidor SQL.

4 Implementação da Solução

O presente capítulo tem como objetivo descrever a implementação da solução para resolver os principais problemas identificados. Para isso, é apresentado um processo detalhado para lidar com as tarefas mais complexas, juntamente com as metodologias utilizadas durante o desenvolvimento. O foco está na descrição da solução implementada, incluindo como ela foi avaliada num contexto real de uso. Com isso, o leitor terá uma compreensão completa de como a solução foi concebida, desenvolvida, implementada e avaliada.

4.1 Descrição da Implementação

Esta secção está dividida de acordo com as principais fases identificadas do projeto. Cada divisão descreve a abordagem utilizada na resolução da mesma e apresenta justificações para o uso de determinado procedimento. Primeiramente será descrita a fase de recolha do *dataset*, seguidamente será descrita a fase de exploração dos dados e construção das sequências, depois são descritos os passos de desenvolvimento dos modelos que envolvem este projeto e, por fim, é apresentado o desenvolvimento da aplicação web para demonstração dos resultados do projeto.

4.1.1 Recolha do Dataset

Sendo o *dataset* um dos pilares para a criação de um modelo de Inteligência Artificial e após ser verificado que, para o caso específico deste projeto, não haveria nenhuma amostra que poderia

ser usada, foi necessária a recolha de dados neste sentido. O *dataset* foi obtido e construído pelo supervisor externo, deste projeto, Luís Costa.

A arquitetura utilizada para esta função está descrita na figura 36:

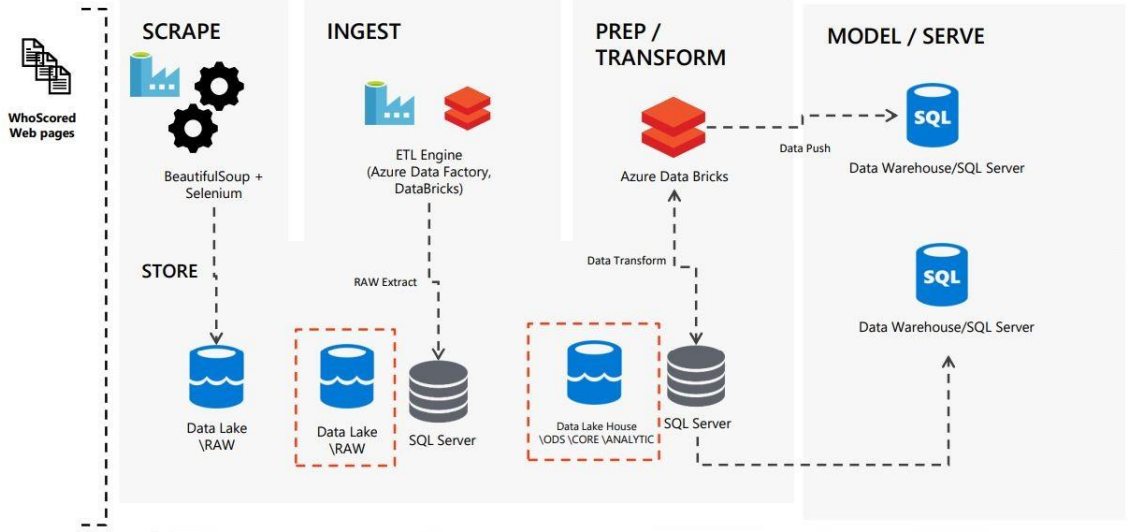


Figura 36: Diagrama de Arquitetura da Lógica de Recolha do *Dataset*

O processo de *scraping* de dados é uma técnica amplamente utilizada para extrair informações valiosas de diferentes fontes na web (ParseHub, 2023). Neste contexto, foi utilizada a ferramenta Selenium para extração dos dados de eventos de futebol da página de resultados desportivos WhoScored. O Selenium é uma ferramenta que permite a automação de navegadores web, ideal para simular a interação humana com uma página web (Selenium, n.d.). Uma vez obtidos os dados, foi necessário armazenar os dados. Neste caso optou-se por utilizar um *data lake*, que é um repositório centralizado de dados, não estruturados e em diferentes formatos, permitindo assim escalabilidade. De seguida, foram estabelecidas pipelines para processar e transformar os dados brutos do *data lake*. Durante este processo de ETL, foram removidos dados duplicados, preenchidos valores ausentes, mapeados valores para os dados seguirem OPTA (OPTA Documentation, n.d.), etc. Após isto, os dados são armazenados em tabelas no SQL Server, para que possam alimentar e servir os modelos de AI deste projeto.

4.1.1.1 *Scraping* e Orquestração

Para a tarefa de *Scraping* e Orquestração são utilizados dois componentes principais: o Raspberry Pi 4 e o Apache Airflow.

O Raspberry Pi 4 é um pequeno computador de placa única criado inicialmente para ensinar ciência da computação básica em escolas e países em desenvolvimento. No entanto, as suas aplicações foram muito além disso. Com o seu baixo custo, modularidade e design aberto, encontrou utilidade em vários domínios. Por exemplo, é amplamente utilizado para monitorização do clima devido à sua acessibilidade e compatibilidade com dispositivos HDMI e USB (CAWLEY, 2022). Assim, este dispositivo eletrónico tornou-se a ferramenta perfeita para as necessidades de *scraping* e orquestração.

Figura 37: Raspberry Pi 4

Em termos de orquestração, foi utilizado o Apache Airflow, uma ferramenta de código aberto projetada para criar, agendar e monitorar fluxos de trabalho de forma programática. É conhecido pela sua robustez e é comumente adotado por engenheiros de dados para simplificar e gerenciar pipelines complexos (Apache, n.d.). Com o Airflow, foi possível visualizar facilmente as dependências, o progresso, os logs, o código e o status de sucesso de todas as *pipelines* de dados.

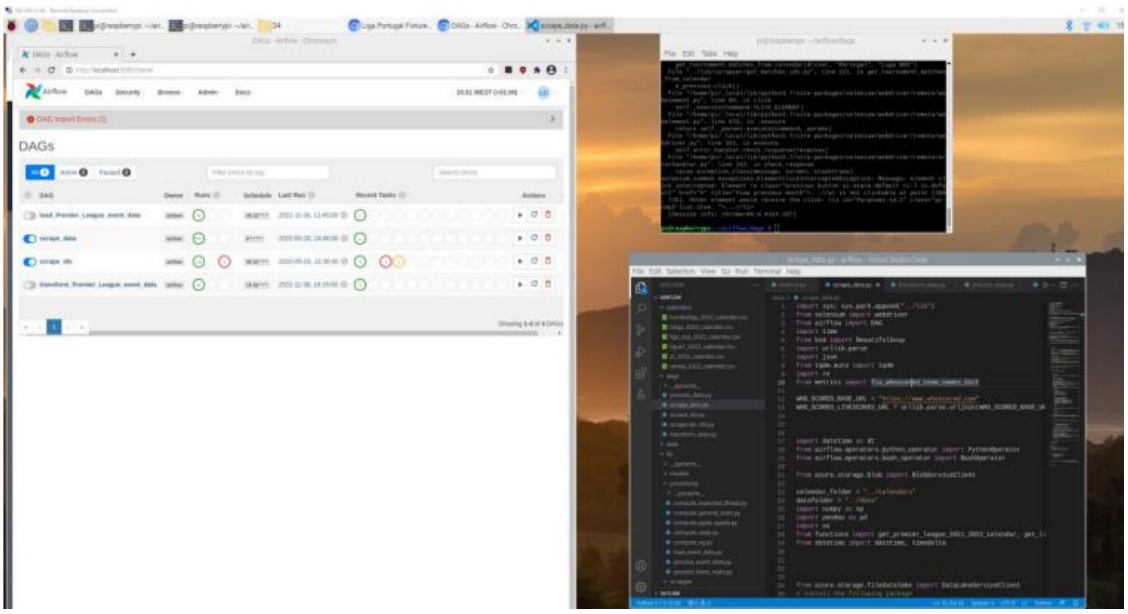


Figura 38: Apache Airflow

Duas *pipelines* foram orquestradas:

- A primeira com o objetivo de fazer *scraping* dos IDs dos jogos. Alguns países, como Portugal, não possuem um calendário pré-definido no início de uma competição. Como o calendário pode mudar devido a vários fatores, como COVID-19, partidas decisivas ou problemas climáticos, é crucial ter um cronograma preciso. Para conseguir isso, foi automatizado um processo para semanalmente extrair os IDs dos eventos e as respetivas datas e horas. Ao executar esta *pipeline* uma vez por semana, é possível garantir que, as datas dos jogos, são as mais atualizadas e precisas possíveis. O resultado deste processo é um ficheiro JSON que contém o calendário completo para os jogos restantes, e o mesmo é armazenado numa Azure Blob Storage;

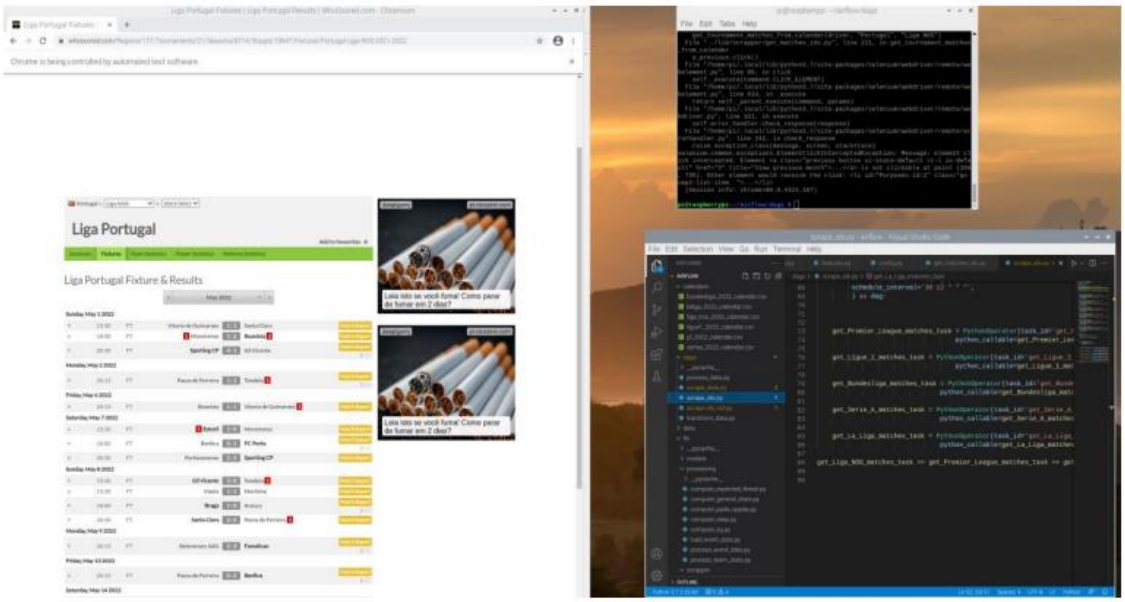


Figura 39: Job de Scraping dos IDs dos jogos

- A segunda, e última, tinha como objetivo a obtenção dos dados em si. Com base no calendário resultante do processo anterior, com as datas e horários dos jogos, este processo de *scraping* de dados é iniciado. Esta *pipeline* concentra-se em fazer *scraping* dos jogos que começaram na última hora ou mais tarde, aquando da inicialização do processo. Este utiliza o HTML que contém os dados necessários para algumas representações visuais. O processo extrai as informações necessárias de cada correspondência e guarda-as como um ficheiro JSON separado. Estes ficheiros são armazenados numa Azure Blob Storage para processamento ou análise posterior.

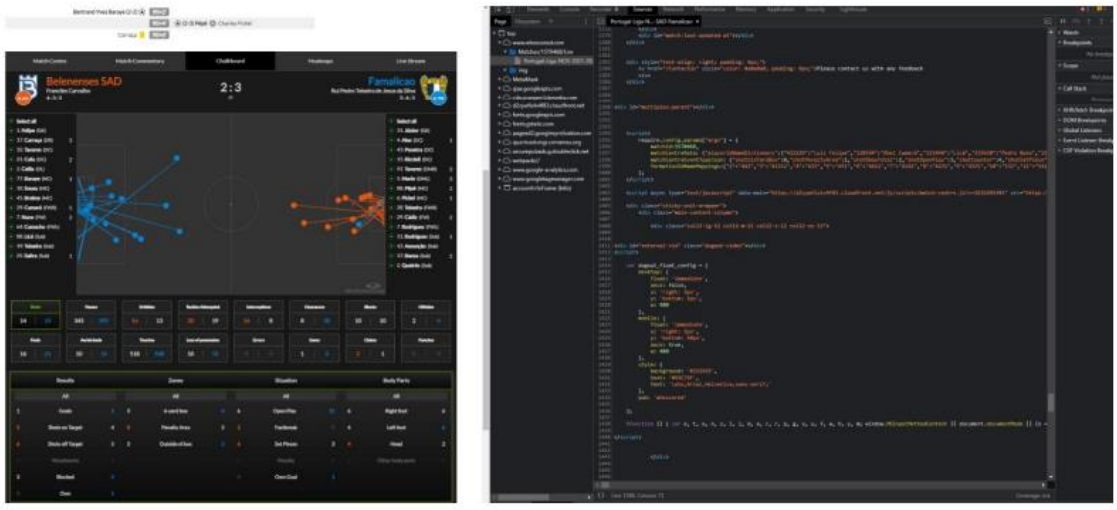


Figura 40: Job de Scraping de Dados dos Jogos

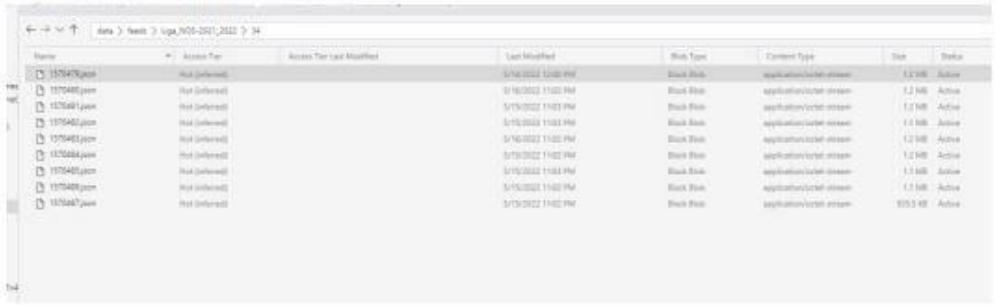


Figura 41: Azure Blob Storage com os ficheiros resultados

4.1.1.2 Ficheiro Resultante

Um exemplo de um ficheiro resultante do processo anterior é um ficheiro JSON que contém mais de 100.000 entradas, cada uma correspondendo a um evento do jogo correspondente. Esses eventos abrangem todas as ações com a bola durante toda a partida, como passes, remates e muito mais. Cada entrada de um evento inclui atributos como ID, minuto, segundo, coordenadas (x, y), ID do jogador, tipo de evento, comprimento e ângulo. Esse nível de granularidade fornece uma visão abrangente de toda a partida de futebol, tornando-o um conjunto de dados valioso e caro. A obtenção desses dados pode custar até 10.000 € por temporada, e empresas como a Opta ou StatsBomb oferecem esses serviços de dados, aplicados a visualizações que permitem inferências valiosas.

Além disso, existe um serviço ainda mais caro, conhecido como dados de rastreamento. Esses dados capturam a posição de cada jogador em campo num determinado período de tempo. Ao contrário dos dados de evento, que fornecem apenas informações sobre o jogador relacionado com o evento, os dados de rastreamento incluem os dados posicionais precisos de todos os jogadores. A obtenção de dados de rastreamento requer algoritmos sofisticados de visão computacional, particularmente técnicas de deteção de objetos. É considerada a mina de ouro de dados para análise do futebol, como já referido no capítulo de Estado de Arte do presente documento.

Figura 42: Exemplo de Ficheiro Resultante

4.1.1.3 Carregamento dos dados

Com a pipeline de scraping a correr, podemos garantir que os dados dos jogos mais recentes são armazenados na Azure Blob Storage. Para ingerir e processar esses dados, foi usado o Azure Databricks para orquestrar o fluxo de ETL, extração, transformação e carregamento.

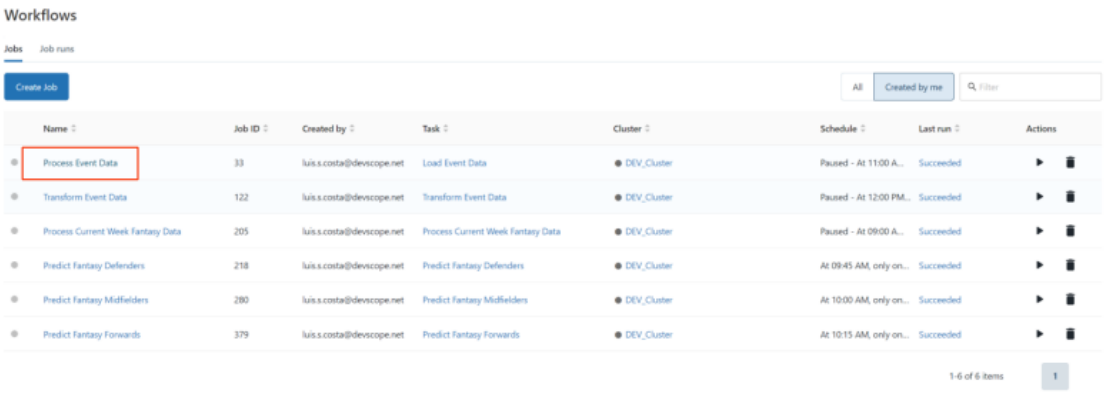
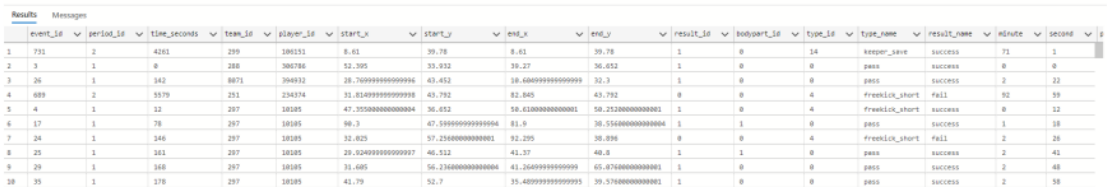


Figura 43: Workflows no Databricks

Para otimizar os custos, o fluxo foi agendado para ser executado diariamente no Databricks. Essa abordagem ajuda a gerenciar o tempo de atividade do cluster e garante a utilização

eficiente de recursos. O fluxo de Carregamento dos dados abrange várias etapas, incluindo ingestão de dados, análise de arquivo JSON, mapeamento de atributos dimensionais, adição de recursos temporais (como tempo em segundos), criação de DataFrames e armazenamento da saída em *delta tables* e SQL Server. Ao seguir esse fluxo de trabalho, é possível processar e transformar os dados extraídos de forma eficaz.



Results Messages

	event_id	period_id	time_seconds	team_id	player_id	start_x	start_y	end_x	end_y	result_id	bodypart_id	type_id	type_name	result_name	minute	second
1	731	2	4261	299	100151	8.61	39.78	8.61	39.78	1	0	14	keeper_save	success	71	1
2	3	1	0	288	306786	52.395	35.932	39.27	36.652	1	0	0	pass	success	0	0
3	26	1	142	8071	394932	28.769999999999996	43.452	10.604999999999999	32.3	1	0	0	pass	success	2	22
4	689	2	5579	251	234374	31.814999999999998	43.792	82.845	43.792	0	0	4	freekick_short	fail	92	59
5	4	1	12	207	10105	47.355000000000004	36.652	50.61000000000001	50.25200000000001	1	0	4	freekick_short	success	0	12
6	17	1	78	207	10105	90.3	47.599999999999994	81.9	38.556000000000004	1	1	0	pass	success	1	18
7	24	1	146	207	10105	32.025	57.25600000000001	92.295	38.896	0	0	4	freekick_short	fail	2	26
8	25	1	161	207	10105	29.924999999999997	46.512	41.37	40.8	1	1	0	pass	success	2	41
9	29	1	168	207	10105	31.605	56.236000000000004	41.264999999999999	65.07600000000001	1	0	0	pass	success	2	48
10	35	1	178	207	10105	41.79	52.7	35.489999999999995	39.57600000000001	1	0	0	pass	success	2	58

Figura 44: Dados resultantes de toda a Pipeline de Carregamento de Dados

O Dataset resultante do fluxo descrito inclui os dados de eventos de 7 temporadas (desde 2016/17 até à temporada atual), das 7 principais ligas europeias (Liga Inglesa, Espanhola, Italiana, Alemã, Francesa, Portuguesa e Holandesa). Isto totaliza um valor aproximado de 40 milhões de dados de eventos, mais precisamente 40 651 646.

4.1.2 Criação das Sequências

Com os dados no SQL Server e prontos para serem utilizados, inicia-se a fase de exploração dos dados para criação de Sequências.

De acordo com a definição de sequência, presente no capítulo de Estado da Arte, foi preciso ter em consideração os seguintes requisitos, para a construção das sequências de um jogo:

- A primeira sequência inicia sempre no primeiro evento do jogo;
- Uma sequência inicia-se com uma ação controlada sobre a bola, ou seja, passe ou ação defensiva controlada;
- Uma sequência é terminada com uma ação defensiva adversária, paragem do jogo, remate ou término dos eventos (final do jogo).

Com isto em mente, foi criada a seguinte função que, passado um conjunto de eventos de um jogo, esta associa um id de sequência a cada evento e a sua numeração dentro da sequência:

```
1.  def extract_sequences(actions: pd.DataFrame) -> pd.DataFrame:
2.      actions_df = actions.copy()
3.      actions_df = actions_df.sort_values(by=[‘game_id’, ‘time_seconds’])
4.      actions_df = actions_df.loc[~(actions_df[‘type_name’].isin([‘challenge’,
    ‘failed_take_on’]))].reset_index(drop=True)
5.      combinations = actions_df[[‘game_id’, ‘team_id’,
    ‘period_id’]].ne(actions_df[[‘game_id’, ‘team_id’, ‘period_id’]].shift())
```

```
6.      sequences = actions_df.groupby((combinations.game_id | combinations.team_id |
    combinations.period_id).cumsum())
7.      actions_df['sequence_number'] = sequences.cumcount().add(1)
8.      actions_df['sequence_id'] = sequences.ngroup().add(1)
9.      actions_df['sequence_id'] = actions_df['game_id'].astype(str) + "-" +
    actions_df['sequence_id'].astype(str)
10.
11.     return actions_df
```

Listagem 1: Função de Extração das Sequências

Como é possível ver pela listagem anterior, inicialmente é feita uma cópia do DataFrame com os dados de eventos para que o DataFrame original permanecesse inalterado. De seguida, os dados são ordenados por jogo e segundos, para garantir que os dados estavam efetivamente pela ordem correta. Depois são retirados os eventos considerados inúteis para esta função, sendo eles o "challenge" e "failed_take_on" que correspondem a tipos de interceção falhadas. Combinações de booleanos são feitas, envolvendo o id do jogo, equipa e período (primeira e segunda parte), ou seja, é verificado se estes valores são iguais aos valores do evento seguinte, caso um deles seja diferente, significa que naquele momento começa uma sequência nova.

As sequências são então agrupadas para que sejam preenchidas as colunas do id de sequência e numeração da ação na sequência. O "sequence_number" corresponde à numeração da ação dentro da sequência, começando por 1 e o id é formado com o id do jogo e numeração da sequência dentro do jogo.

Estas sequências são o pilar deste projeto e utilizadas em todas as fases do mesmo.

4.1.3 Modelo de xG

A construção de um modelo de Expected Goals (xG) envolve diversas etapas que visam o objetivo de estimar a probabilidade de um determinado evento resultar num golo. Estas fases incluem a identificação das características relevantes para o cálculo do mesmo, criação de um conjunto de treino, validação e teste, construção do modelo, treino e teste deste, avaliação do desempenho e afinação para que este tenha os melhores resultados possíveis. Estas fases são descritas abaixo.

4.1.3.1 Construção das *Features* (Características)

Esta primeira etapa de implementação envolveu a compreensão do conjunto de dados disponíveis, identificação das variáveis relevantes e criação das *features* relevantes que seriam usadas no modelo de xG, baseado na pesquisa executada para escrita do Estado da Arte.

Com este objetivo em mente, em primeiro lugar, são filtrados os eventos pertencentes ao tipo de ações que são consideradas para o cálculo do xG e ordenados os eventos de acordo com o jogo, período e tempo em segundos. De seguida, contrariamente ao que foi averiguado na maioria dos exemplos de modelos de xG encontrados, foi definida uma função que cria e retorna um conjunto de listas compostas pelas n ações anteriores, com o objetivo de averiguar a influência das ações anteriores nas atuais.

```
1.  def gamestates (actions: pd.DataFrame, nb_prev_actions: int = 3) -> GameStates:
2.      states = [actions]
3.      for i in range(1, nb_prev_actions):
4.          prev_actions = actions.copy().shift(i, fill_value=0)
5.          prev_actions.loc[: i-1, :] = pd.concat([actions[:1]] * i,
    ignore_index=True)
6.          states.append(prev_actions)
7.      return states
8.
```

Listagem 2: Função de Retorno das n ações anteriores

Neste caso foram escolhidas as 3 ações anteriores e um objeto de listas de listas foi retornado. Cada lista era então composta pela ação e acompanhada pelas 3 ações anteriores. Após isto foram calculadas e aplicadas as seguintes características:

- **Tipo de ação**, foi utilizado one-hot encoding sobre os tipos de ações. Alguns tipos de ação incluem: passe, cruzamento, falta, corte, etc.
- Se a ação consistia numa "***big change***", grande chance de golo;
- **Parte do Corpo**, foi utilizado one-hot encoding sobre cada parte do corpo possível. São exemplos: pé, cabeça, outro, etc.
- **Direção do movimento da ação**, através das coordenadas de início, fim da ação e ângulo da ação, foi calculada a direção do movimento da ação;
- **Diferença de Golos**, calculado subtraindo o número de golos marcados pelo adversário pelo número de golos da equipa que preformou a última ação. Este valor será positivo se a equipa atual estiver em vantagem de golos, negativo se estiver em desvantagem;
- **Localização** de início, conjunto das coordenadas de início da ação;

- **Localização** de fim, conjunto das coordenadas de fim da ação;
- **Jogador** que performou a ação, corresponde ao id do jogador que fez a ação;
- **Resultado da ação**, foi feito one-hot encoding sobre os diversos tipos de resultados. Alguns exemplos são: sucesso, falhanço, fora de jogo, falta, etc.
- **Espaço percorrido**, corresponde ao espaço percorrido durante a ação, calculado a partir da coordenada inicial e coordenada final;
- **Equipa**, corresponde ao id da equipa que performa a ação;
- Existência de "**through ball**", verifica se a ação consiste numa enfiada de bola ou passe em desmarcação;
- **Tempo de Jogo**, diz respeito aos segundos desde o começo do jogo;
- **Intervalo de Tempo**, corresponde à diferença entre as n ações anteriores e a ação atual;
- **Tipo de Jogada**, one-hot encoding dos diversos tipos de jogada existentes, como: contra-ataque, jogada de canto, pontapé de falta, etc.

Para além disto, foram ainda acrescentadas duas outras características, já calculadas, mas noutra unidade de medida:

- **Coordenadas Polares**, utilizado a distância para o golo e o ângulo para o golo;
- **Direção do movimento em coordenadas polares**. À semelhança da direção normal, esta característica foi calculada, convertendo as mesmas componentes utilizadas para o cálculo anterior, para unidades polares.

De seguida, as funções dos cálculos destas características foram aplicadas às n ações obtidas anteriormente e o resultado foi um DataFrame contendo todas as características com um sufixo "_ai", sendo que i correspondia ao número da ação. Tendo como exemplo a primeira característica enumerada, "type_pass_a0", correspondia ao one-hot encoding do tipo da ação da ação atual corresponder a um passe, "type_shot_a2" correspondia à antepenúltima ação relativamente à atual, ser um remate.

Após isto, foram averiguados os índices das ações que correspondiam a remates, da listagem de ações inicial, a que foi passada à função de criação dos "gameStates" e filtrado o DataFrame de características para apenas remates. Este conjunto de ações foi necessário para que inicialmente, todos os tipos de ações fossem utilizadas para cálculo dos "gameStates", mas para passagem ao modelo, interessavam apenas ações atuais de remates. Por outras palavras, neste momento teríamos todas as características de remates e das 3 ações anteriores.

4.1.3.2 Construção e Treino do Modelo

Os dados relativos às temporadas desde 2016/17 até à época passada (2021/22), das principais 6 ligas europeias, foram utilizados para treino e validação. A temporada atual (2022/23) foi escolhida para ser usada como teste.

As colunas relativas a 'result_success_a0', 'result_fail_a0', 'result_owngoal_a0', foram retiradas do conjunto de características pois já apresentavam os resultados que iriam ser calculados pelo modelo. A coluna 'result_success_a0' foi utilizada como característica objetivo.

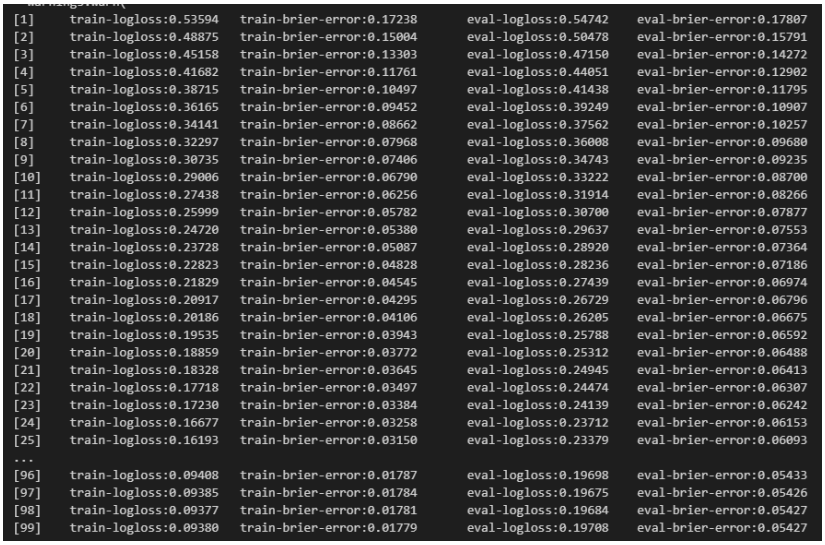
```
1.  X = shot_actions.drop(columns=['result_success_a0', 'result_fail_a0',
    'result_owngoal_a0'])
2.  y = shot_actions.result_success_a0
3.
4.  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)
5.
```

Listagem 3: Divisão do Dataset utilizando Holdout

A técnica Holdout foi utilizada para divisão do conjunto de dados de treino e validação, 80% de treino e 20% para validação.

Foi escolhida a métrica Brier Score Error (BSE) para avaliar a qualidade das previsões do modelo, pois esta mede a discrepância entre as probabilidades previstas por modelos e ocorrências reais de eventos. Um BSE baixo indica que as probabilidades estão bem calibradas, ou seja, próximas da taxa real de ocorrência de eventos, contrariamente, um BSE alto, indica que as probabilidades estão descalibradas (SKLearn Metrics, n.d.).

O modelo escolhido foi o xgboost e, para tal, foi necessário a criação de matrizes para os dados serem passados ao modelo. Foram utilizados os parâmetros *default* e foram corridas 100 épocas.



```
[1]   train-logloss:0.53594   train-brier-error:0.17238   eval-logloss:0.54742   eval-brier-error:0.17807
[2]   train-logloss:0.48875   train-brier-error:0.15004   eval-logloss:0.50478   eval-brier-error:0.15791
[3]   train-logloss:0.45158   train-brier-error:0.13303   eval-logloss:0.47150   eval-brier-error:0.14272
[4]   train-logloss:0.41682   train-brier-error:0.11761   eval-logloss:0.44051   eval-brier-error:0.12902
[5]   train-logloss:0.38715   train-brier-error:0.10497   eval-logloss:0.41438   eval-brier-error:0.11795
[6]   train-logloss:0.36165   train-brier-error:0.09452   eval-logloss:0.39249   eval-brier-error:0.10907
[7]   train-logloss:0.34141   train-brier-error:0.08662   eval-logloss:0.37562   eval-brier-error:0.10257
[8]   train-logloss:0.32297   train-brier-error:0.07968   eval-logloss:0.36008   eval-brier-error:0.09680
[9]   train-logloss:0.30735   train-brier-error:0.07406   eval-logloss:0.34743   eval-brier-error:0.09235
[10]  train-logloss:0.29006   train-brier-error:0.06790   eval-logloss:0.33222   eval-brier-error:0.08700
[11]  train-logloss:0.27438   train-brier-error:0.06256   eval-logloss:0.31914   eval-brier-error:0.08266
[12]  train-logloss:0.25999   train-brier-error:0.05782   eval-logloss:0.30700   eval-brier-error:0.07877
[13]  train-logloss:0.24720   train-brier-error:0.05380   eval-logloss:0.29637   eval-brier-error:0.07553
[14]  train-logloss:0.23728   train-brier-error:0.05087   eval-logloss:0.28920   eval-brier-error:0.07364
[15]  train-logloss:0.22823   train-brier-error:0.04828   eval-logloss:0.28236   eval-brier-error:0.07186
[16]  train-logloss:0.21829   train-brier-error:0.04545   eval-logloss:0.27439   eval-brier-error:0.06974
[17]  train-logloss:0.20917   train-brier-error:0.04295   eval-logloss:0.26729   eval-brier-error:0.06796
[18]  train-logloss:0.20186   train-brier-error:0.04106   eval-logloss:0.26205   eval-brier-error:0.06675
[19]  train-logloss:0.19535   train-brier-error:0.03943   eval-logloss:0.25788   eval-brier-error:0.06592
[20]  train-logloss:0.18859   train-brier-error:0.03772   eval-logloss:0.25312   eval-brier-error:0.06488
[21]  train-logloss:0.18328   train-brier-error:0.03645   eval-logloss:0.24945   eval-brier-error:0.06413
[22]  train-logloss:0.17718   train-brier-error:0.03497   eval-logloss:0.24474   eval-brier-error:0.06307
[23]  train-logloss:0.17230   train-brier-error:0.03384   eval-logloss:0.24139   eval-brier-error:0.06242
[24]  train-logloss:0.16677   train-brier-error:0.03258   eval-logloss:0.23712   eval-brier-error:0.06153
[25]  train-logloss:0.16193   train-brier-error:0.03150   eval-logloss:0.23379   eval-brier-error:0.06093
...
[96]  train-logloss:0.09408   train-brier-error:0.01787   eval-logloss:0.19698   eval-brier-error:0.05433
[97]  train-logloss:0.09385   train-brier-error:0.01784   eval-logloss:0.19675   eval-brier-error:0.05426
[98]  train-logloss:0.09377   train-brier-error:0.01781   eval-logloss:0.19684   eval-brier-error:0.05427
[99]  train-logloss:0.09380   train-brier-error:0.01779   eval-logloss:0.19708   eval-brier-error:0.05427
```

Figura 45: Treino do Modelo de xG

Foram ainda calculados os valores de xG para casos especiais como os penaltis e as recargas. Para o valor dos penaltis foi utilizado o valor de 0.775 pois foi calculada a média dos penaltis convertidos em golo no *dataset* de treino. Mais informação em relação ao cálculo deste tipo de casos especiais pode ser encontrada no capítulo 2 do presente documento.

```
1.  def calculate_penalty_xg(shots):
2.      return np.where(shots.type_name == 'shot_penalty', 0.775, shots.xG).round(5)
3.
4.
5.  def calculate_rebound_xg(shots):
6.      for i, shot in enumerate(shots):
7.          if (i==0): continue
8.          delta_time = shots.iloc[i, shots.columns.get_loc('time_seconds')] –
    shots.iloc[i-1, shots.columns.get_loc('time_seconds')]
9.          if (shots.iloc[1, shots.columns.get_loc('period_id')] == shots.iloc[i-1,
    shots.columns.get_loc('period_id')] and delta_time < 5):
10.             shots.iloc[i, shots.columns.get_loc('xG')] = (1- ((1 – shots.iloc[i-1,
    shots.columns.get_loc('xG')]) * (1 – shots.iloc[i, shots.columns.get_loc('xG')])))
11.     return shots.xG
12.
```

Listagem 4: Funções de Cálculo de xG para Penaltis e Recargas

4.1.3.3 Afinação do Modelo

Com o objetivo de melhorar o modelo construído no passo anterior, foram aplicadas duas técnicas, o cálculo da importância de cada característica (*Feature Importance*) e o ajuste através de Hiperparâmetros (HyperParameter Tuning). A primeira era necessária para remover as características menos relevantes, de modo a criar um modelo mais eficaz e menos complexo. A

segunda, consistia no ajuste de diversos valores que podem ser passados ao modelo, de modo a encontrar a combinação que maximiza as métricas de avaliação do modelo.

Após o cálculo da importância das características, foi obtido o seguinte resultado:

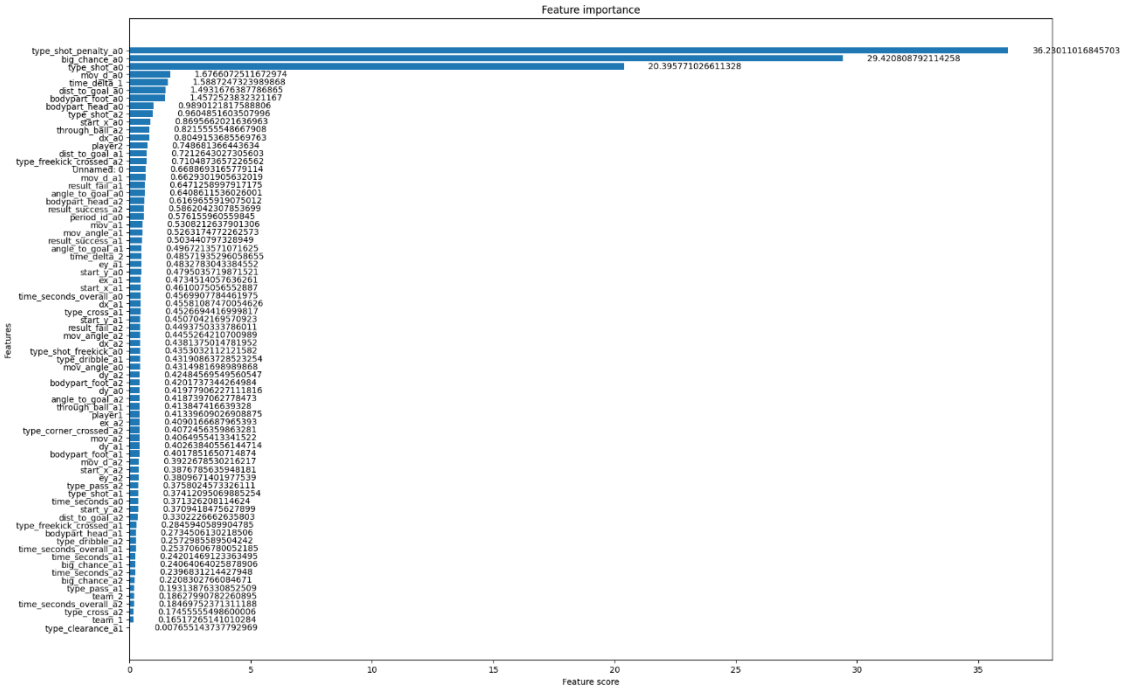


Figura 46: Gráfico de Feature Importance

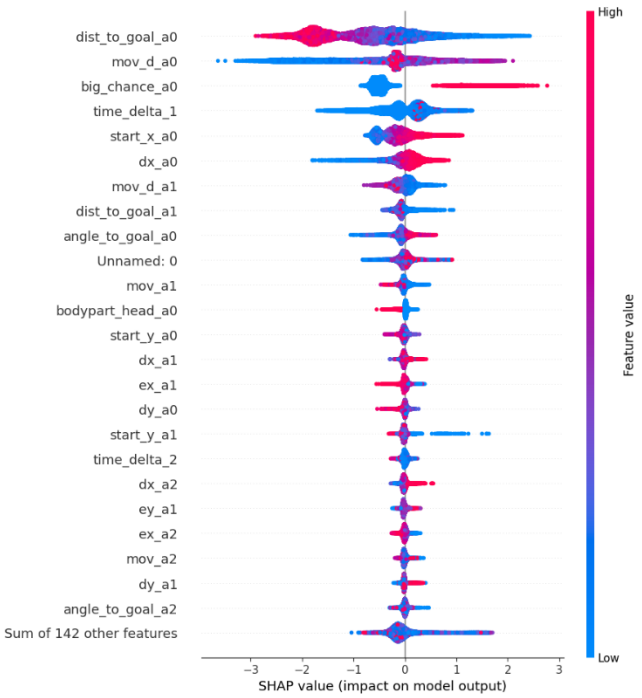


Figura 47: Gráfico de Beeswarm de Feature Importance

As características menos utilizadas, neste caso com um valor inferior a 0.3, foram removidas.

Em relação aos ajustes dos Hiperparâmetros, este foi efetuado com o auxílio da biblioteca hyperopt. O espaço de hiperparâmetros incluía:

```
1.  space = {'max_depth': scope.int(hp.quniform('max_depth', 1, 30, 2)),
2.           'learning_rate': scope.float(hp.quniform('learning_rate', 0.01, 0.5,
    0.025)),
3.           'eta': scope.float(hp.uniform('eta', 0.001, 0.5)),
4.           'subsample': hp.quniform('subsample', 0.3, 1, 0.01),
5.           'colsample_bylevel': scope.float(hp.quniform('colsample_bylevel', 0.1,
    1.0, 0.01)),
6.           'colsample_bytree': scope.float(hp.quniform('colsample_bytree', 0.1, 1.0,
    0.01))}
```

Listagem 5: Espaço de Hiper Parâmetros

Os melhores parâmetros foram então utilizados e a apresentação e avaliação dos resultados pode ser encontrada no subcapítulo 4.2 do presente documento.

4.1.4 Modelo de xT

O desenvolvimento do modelo de xT, similarmente ao anterior, envolveu duas etapas essenciais. Estas fases incluem o treino do modelo que envolve o cálculo de diversas matrizes e probabilidades e, por fim, a construção da pipeline de previsão.

4.1.4.1 Treino do Modelo

O treino do modelo envolveu o cálculo de várias probabilidades relacionadas com as ações ocorridas durante o jogo. Para isto, foram utilizadas as ações das principais 6 ligas europeias, desde a temporada 2016 até à temporada passada.

O processo de treino consistiu nos seguintes passos:

Cálculo da matriz de probabilidade de marcar golo (scoring_prob_matrix): A partir das ações de remate realizadas durante o jogo, são contabilizados quantos remates ocorreram em cada célula da grade de jogo. Em seguida, é calculada a quantidade de golos marcados em cada célula.

```
1.  def scoring_prob(actions: DataFrame, l: int = N, w: int = M) -> np.ndarray:
2.
3.      shot_actions = actions[(actions.type_name == 'shot')]
4.      goals = shot_actions[(shot_actions.result_name == 'success')]
5.
```

```
6.      shotmatrix = _count(shot_actions.start_x, shot_actions.start_y, l, w)
7.      goalmatrix = _count(goals.start_x, goals.start_y, l, w)
8.      return _safe_divide(goalmatrix, shotmatrix)
```

Listagem 6: Função de Cálculo da Matriz de Probabilidade de Golo

A matriz de probabilidade de marcar golo é obtida dividindo a matriz de golos marcados pela matriz de remates realizados.

Cálculo das matrizes de probabilidade de escolher rematar (shot_prob_matrix) **e de escolher se mover** (move_prob_matrix): As ações de progressão da bola, como passes, dribles e cruzamentos, são selecionadas. A partir dessas ações, são contabilizadas quantas vezes ocorreu uma ação de remate e quantas vezes ocorreu uma ação de progressão da bola.

```
1.  def action_prob(
2.      actions: DataFrame, l: int = N, w: int = M
3.  ) -> Tuple[np.ndarray, np.ndarray]:
4.
5.      move_actions = get_move_actions(actions)
6.      shot_actions = actions[(actions.type_name == 'shot')]
7.
8.      movematrix = _count(move_actions.start_x, move_actions.start_y, l, w)
9.      shotmatrix = _count(shot_actions.start_x, shot_actions.start_y, l, w)
10.     totalmatrix = movematrix + shotmatrix
11.
12.     return _safe_divide(shotmatrix, totalmatrix), _safe_divide(movematrix,
    totalmatrix)
```

Listagem 7: Função de Cálculo das Matrizes de Probabilidade de escolher Rematar e de escolher se mover

As matrizes de probabilidade de escolher rematar e de escolher se mover são obtidas dividindo as contagens pelas respetivas contagens totais.

Cálculo da matriz de transição de movimento (transition_matrix): A partir das ações de progressão da bola, são determinadas as células de início e fim de cada movimento. A matriz de transição de movimento é calculada com base nas probabilidades de sucesso de cada movimento entre as células.

```
1.  def move_transition_matrix(actions: DataFrame, l: int = N, w: int = M) ->
    np.ndarray:
2.
3.      move_actions = get_move_actions(actions)
4.
5.      X = pd.DataFrame()
6.      X['start_cell'] = _get_flat_indexes(move_actions.start_x,
    move_actions.start_y, l, w)
7.      X['end_cell'] = _get_flat_indexes(move_actions.end_x, move_actions.end_y, l,
    w)
8.      X['result_name'] = move_actions.result_name
```

```
9.
10.     vc = X.start_cell.value_counts(sort=False)
11.     start_counts = np.zeros(w * l)
12.     start_counts[vc.index] = vc
13.
14.     transition_matrix = np.zeros((w * l, w * l))
15.
16.     for i in range(0, w * l):
17.         vc2 = X[((X.start_cell == i) & (X.result_name ==
    'success'))].end_cell.value_counts(
18.             sort=False
19.         )
20.         transition_matrix[i, vc2.index] = vc2 / start_counts[i]
21.
22.     return transition_matrix
```

Listagem 8: Função de Cálculo da Matriz de Transição de Movimento

Resolução da equação de xT criada por Karun Singh. A cada iteração, é calculada uma matriz de pagamento total para cada célula, que leva em consideração a probabilidade de marcar golo, a probabilidade de escolher se mover e a matriz de transição de movimento. O processo é repetido até que a diferença entre as iterações subsequentes seja menor que uma tolerância especificada. A tolerância escolhida foi $1e^{-5}$.

```
1.  scoring_prob_matrix = scoring_prob(actions, self.l, self.w)
2.  shot_prob_matrix, self.move_prob_matrix = action_prob(actions, self.l, self.w)
3.  transition_matrix = move_transition_matrix(actions, self.l, self.w)
4.  gs = p_scoring * p_shot
5.  diff = 1
6.  it = 0
7.  self.heatmaps.append(self.xT.copy())
8.
9.  while np.any(diff > self.eps):
10.     total_payoff = np.zeros((self.w, self.l))
11.
12.     for y in range(0, self.w):
13.          for x in range(0, self.l):
14.               for q in range(0, self.w):
15.                    for z in range(0, self.l):
16.                         total_payoff[y, x] += (
17.                             transition_matrix[self.l * y + x, self.l * q + z] *
    self.xT[q, z]
18.                         )
19.
20.     newxT = gs + (p_move * total_payoff)
21.     diff = newxT - self.xT
22.     self.xT = newxT
23.     self.heatmaps.append(self.xT.copy())
24.     it += 1
```

Listagem 9: Treino do Modelo

4.1.4.2 Pipeline de Previsão

Depois do treino e de serem calculados os valores relativos aos pesos de cada célula da grelha, foi construída a *pipeline* para uso do modelo.

Inicialmente, foi realizada uma fase de preparação dos dados para serem passados ao modelo. Neste caso foi necessário a filtragem das ações realizadas com sucesso, para os tipos desejados:

```
25. def get_successful_move_actions(actions: DataFrame) -> DataFrame:
26.      move_actions = actions[
27.         (actions.type_name == "pass")
28.         | (actions.type_name == "dribble")
29.         | (actions.type_name == "cross")
30.         | (actions.type_name == "throw_in")
31.     ]
32.      Return move_actions[move_actions.result_name == 'success']
33.
```

Listagem 10: Função de Retorno das ações bem-sucedidas para o xT

Aqui apenas era interessante o cálculo para os tipos de ações averiguados em cima.

Depois de serem passadas as ações para o modelo, este calcula as células de inicio e fim da ação e, conforme a matriz de valores xT calculada no treino, é calculada a diferença entre o perigo calculado na célula de fim e de início e retornado o valor.

```
1.  startxc, startyc = _get_cell_indexes(actions.start_x, actions.start_y, l, w)
2.  endxc, endyc = _get_cell_indexes(actions.end_x, actions.end_y, l, w)
3.
4.  xT_start = grid[w - 1 - startyc, startxc]
5.  xT_end = grid[w - 1 - endyc, endxc]
6.
7.  return xT_end - xT_start
```

Listagem 11: Trecho de código do cálculo de xT

Em resumo, a construção deste modelo de xT segue os princípios e o cálculo proposto por Karun Singh.

4.1.5 Estatísticas de Ações, Sequências e Equipas

Depois da obtenção das estatísticas avançadas dos xG e xT, achou-se de extrema importância a utilização das propriedades das ações para a criação de mais estatísticas para apresentar na aplicação web final e aplicá-las às sequências e Equipas.

Começando pelo nível de granularidade mais baixo (ação), foram calculadas uma série de estatísticas que viriam a ser usadas como base nos cálculos dos níveis mais acima (sequência e Equipa). Com isto em mente, foram calculadas as seguintes estatísticas:

- **Existência de Passe Longo**. Uma ação consistia num passe longo caso fosse do tipo passe, superior ou igual a 32 metros e o ângulo esteja entre 0 e π/3 ou 5π/3 e 2π;
- **Existência de Passe Longo Vertical**. Semelhante à lógica anterior, seria uma ação do tipo passe, superior ou igual a 32 metros e ângulo, neste caso, esteja 0 e π/4 ou 7π/4 e 2π;
- **Zona da ação relativamente ao estilo de Jogo**. Caso a coordenada X corresponda a um valor inferior a 60 metros, o estilo corresponde a "Maintenance" (Manutenção), se estiver entre 52.5 m e 87.15, ação corresponderia a "Buildup" (Construção) e se for superior a 70m, indica um estilo de jogo de ameaça sustentada ("Sustained Threat"). Uma ação pode corresponder a mais do que um estilo de jogo;
- **Tipo de Jogada**. Atribuição de um tipo de jogada, indo desde bola corrida, contra-ataque, jogada de canto até não ter nenhum tipo associado.

```
1.  def improve_actions(actions: pd.DataFrame) -> pd.DataFrame:
2.      actions_temp = actions.copy()
3.      actions['forward_long_pass'] = calculate_forward_long_passes(actions_temp)
4.      actions['forward_vertical_long_pass'] =
    calculate_forward_vertical_long_passes(actions_temp)
5.      actions['maintenance_style_zone'] = get_maintenance_style_zone(actions_temp)
6.      actions['buildup_style_zone'] = get_buildup_style_zone(actions_temp)
7.      actions['sustained_threat_style_zone'] =
    get_sustained_threat_style_zone(actions_temp)
8.      actions['type_of_play'] = get_type_of_play(actions_temp)
9.      return actions
10.
```

Listagem 12: Função de Cálculo das Estatísticas de Ações

Seguindo o fluxo de granularidade, depois de obtidas as estatísticas das ações das sequências, estas podem ser usadas para criação de estatísticas relativas às próprias sequências. Foram essas:

- **Duração**: obtida da subtração do tempo da última ação pelo tempo da primeira ação;
- **Espaço Percorrido**: obtido através da soma de todas as distâncias percorridas pelas ações;
- **Progressão no Campo**: corresponde à subtração da Coordenada X da ação final pela Coordenada X da ação inicial. Se este valor for positivo significa que a ação contribuiu

positivamente para avançar no terreno, se for negativo, significa que este fim numa posição mais recuada do terreno;

- **Número de Passes**: consiste na soma das ações do tipo passe;
- **Tamanho**: corresponde ao número de ações da sequência;
- **Tipo de Jogada**: corresponde ao tipo de jogada, da última ação da sequência;
- **Número de Jogadores Envolvidos**: corresponde ao número de ids únicos presentes na sequência;
- **Existência de Passe Longo**: verificação da existência de Passe longo nalguma ação da Sequência;
- **Tempo até Passe Longo**: na eventualidade de existência de passe longo, é calculado o tempo entre a ação inicial da sequência e a ação que corresponde ao passe longo;
- **Existência de Passe Longo Vertical**: identicamente à de passe longo normal, verificação de passe longo vertical em alguma ação da sequência;
- **Tempo até Passe Longo Vertical**: mesma lógica que o tempo até passe longo, mas aplicado ao passe longo vertical;
- **Existência de "*Big Chance*", existência de Passe chave (*KeyPass*), existência de "*Through Ball*"**: são 3 estatísticas calculadas a partir da verificação da presença da estatística especifica numa ação da sequência;
- **Zona inicial** e **Zona Final**: zona Defensiva corresponde à Coordenada X entre os valores 0 e 40 metros, Zona de Meio Campo entre 40 e 70m e Zona Atacante dos 70 metros adiante. Para a zona inicial são usados os valores da primeira ação e da zona final, os da última ação;
- **Percentagem de Pertença do estilo de Manutenção** (Maintenance), **Percentagem de Pertença do estilo de Construção** (Buildup) e **Percentagem de Pertença do estilo de Ameaça Sustentada** (Sustained Threat): é calculado o tempo que determinado estilo ocupa na sequência e atribuída uma percentagem para esse valor comparativamente ao tempo total da sequência;
- **Tempo até Chegar ao Terço Final**: para este cálculo foi verificado o tempo desde a ação inicial até uma ação que se passe numa zona de Ameaça Sustentada, caso não aconteça, o valor irá ser nulo;
- Verificar se **acabou em Remate**: é verificado se a última ação da sequência corresponde a uma ação do tipo Remate;

- Verificar se **acabou em Golo**: similarmente à anterior é verificado se a última ação da sequência corresponde a um remate e adicionalmente é também verificado se o resultado da ação foi sucesso;
- **Golo Esperado** (**xG**): corresponde à soma dos valores de xG das ações que constituem a sequência;
- **Ameaça Esperada** (**xT**): é calculada pela divisão da soma dos valores de xT e o número de ações que possuíam xT.

```
1.  def build_sequence_stats(actions: pd.DataFrame) -> pd.DataFrame:
2.      stats = pd.DataFrame()
3.      actions_temp = actions.copy()
4.      stats[‘sequence_id’] = get_sequence_ids(actions_temp)
5.      stats[‘game_id’] = get_sequence_game_ids(actions_temp)
6.      stats[‘team_id’] = get_sequence_team_ids(actions_temp)
7.      stats[‘team_name’] = get_sequence_team_names(actions_temp)
8.      stats[‘length’] = get_sequence_length(actions_temp)
9.      stats[‘type_of_play’] = get_sequence_type_of_play(actions_temp)
10.     stats[‘number_players_involved’] =
    get_sequence_number_players_involved(actions_temp)
11.     stats[‘has_forward_long_pass’] = get_sequence_has_forward_long_pass
    (actions_temp)
12.     stats[‘time_to_forward_long_pass’] = get_sequence_time_to_forward_long_pass
    (actions_temp)
13.     stats[‘has_forward_vertical_long_pass’] =
    get_sequence_has_forward_vertical_long_pass(actions_temp)
14.     stats[‘time_to_forward_vertical_long_pass’] =
    get_sequence_time_to_forward_vertical_long_pass(actions_temp)
15.     stats[‘time_to_reach_final_third’] =
    get_sequence_time_to_reach_final_third(actions_temp)
16.     stats[‘duration_seconds’] = get_sequence_duration(actions_temp)
17.     stats[‘space_covered’] = get_sequence_space_covered(actions_temp)
18.     stats[‘pitch_progression’] = get_sequence_pitch_progression(actions_temp)
19.     stats[‘nmr_passes’] = get_sequence_nmr_passes(actions_temp)
20.     stats[‘has_big_chance’] = get_sequence_has_big_chance(actions_temp)
21.     stats[‘has_keypass’] = get_sequence_has_keypass(actions_temp)
22.     stats[‘has_through_ball’] = get_sequence_has_through_ball(actions_temp)
23.     stats[‘initial_zone’] = get_sequence_initial_zone(actions_temp)
24.     stats[‘final_zone’] = get_sequence_final_zone(actions_temp)
25.     stats[‘maintenance_membership’] =
    get_sequence_maintenance_membership(actions_temp)
26.     stats[‘buildup_membership’] = get_sequence_buildup_membership(actions_temp)
27.     stats[‘sustained_threat_membership’] =
    get_sequence_sustained_threat_membership(actions_temp)
28.     stats[‘ended_in_shot’] = get_sequence_ended_in_shot(actions_temp)
29.     stats[‘ended_in_goal’] = get_sequence_ended_in_goal(actions_temp)
30.     stats[‘xG’] = calculate_xG(actions_temp)
31.     stats[‘xT’] = calculate_xT(actions_temp)
32.
33.
34.     return stats
```

Listagem 13: Função que calcula as Estatísticas relativamente à Sequência

Para finalizar, no que toca às estatísticas do nível mais superficial, foram consideradas sequências de 2 ou mais passes, pois considerou-se que sequências de menos de 2 passes, não

tinham representatividade para as equipas. Com isto em mente, efetuou-se a filtragem e depois verificou-se a média dos valores das sequências, agrupados pela equipa em questão.

```
1.  grouped_df = df_league.groupby('team_id').agg({
2.      'team_id': 'first',
3.      'team_name': lambda x: x.value_counts().index[0],
4.      'number_players_involved': 'mean',
5.      'has_forward_long_pass': lambda x: sum(x) / len(x),
6.      'has_forward_vertical_long_pass': lambda x: sum(x) / len(x),
7.      'time_to_reach_final_third': 'mean',
8.      'duration_seconds': 'mean',
9.      'length': 'mean',
10.     'space_covered': 'mean',
11.     'pitch_progression': 'mean',
12.     'nmr_passes': 'mean',
13.     'maintenance_membership': lambda x: sum(x) / len(x),
14.     'buildup_membership': lambda x: sum(x) / len(x),
15.     'sustained_threat_membership': lambda x: sum(x) / len(x),
16.     'has_big_chance': lambda x: sum(x) / len(x),
17.     'has_keypass': lambda x: sum(x) / len(x),
18.     'has_through_ball': lambda x: sum(x) / len(x),
19.     'initial_zone': lambda x: x.value_counts().index[0],
20.     'final_zone': lambda x: x.value_counts().index[0],
21.     'ended_in_shot': lambda x: sum(x) / len(x),
22.     'ended_in_goal': lambda x: sum(x) / len(x),
23.     'xG': 'mean',
24.     'xT': 'mean',
25. })
```

Listagem 14: Função Criação Estatísticas de Equipas

O conjunto de todas estas estatísticas conseguem transmitir uma representatividade da equipa e irão ser apresentados na página de detalhes da Equipa na aplicação Web.

4.1.6 Autoencoder

O último modelo construído tinha como objetivo perceber as características das sequências e averiguar as semelhanças entre as mesmas, para que associando as sequências a uma equipa, seja possível avaliar as semelhanças nos estilos de jogo das equipas. Todos os passos até à obtenção deste objetivo serão descritos abaixo.

4.1.6.1 Construção do *Dataset*

Com o objetivo descrito acima em mente, foi necessária a criação da representação visual da sequência para alimentar o modelo. Decidiu-se a construção de imagens de 128x128 pixéis e de 256x256 pixéis para, posteriormente, serem comparados os resultados dos modelos destes dois tamanhos e uso do melhor.

Para a produção das imagens, foram utilizadas as coordenadas de inicio e fim das ações das sequências e foi averiguado que era necessário o cálculo das coordenadas de receção da bola. Isto tornou-se necessário pois verificou-se que o início da ação seguinte não era necessariamente a mesma coordenada do fim da ação atual. Assim, atribuiu-se o valor da coordenada final da ação anterior para a coordenada de receção da ação atual:

```
1.  sequences_match['ball_receiving_x'] = np.where(sequences_match.shift(-
    1).result_name.eq('success') & sequences_match.team_id.eq(
    sequences_match.team_id.shift(-1)), sequences_match.end_x.shift(-1), np.nan)
2.
3.  sequences_match['ball_receiving_y'] = np.where(sequences_match.shift(-
    1).result_name.eq('success') &
    sequences_match.team_id.eq(sequences_match.team_id.shift(-1)),
    sequences_match.end_y.shift(-1), np.nan)
4.
```

Listagem 15: Cálculo das Coordenadas de Recepção

Após isto, os valores foram normalizados para os valores do tamanho das imagens (128 ou 256). Foram descartadas as sequências de menos de 2 ações. De seguida, com o auxilio da biblioteca OpenCV foram desenhadas e guardadas as imagens. A listagem que representa a construção da imagem pode ser encontrada em anexo.

Figura 48: Exemplos das Imagens Construídas

Foram construídas as sequências para todas as principais 6 ligas europeias, desde a temporada 2016/17 até à atual.

4.1.6.2 Construção do Modelo

Identicamente ao método utilizado nos modelos anteriores, foram utilizadas para treino e validação, as temporadas passadas e para teste a atual e foi utilizado o método Holdout para divisão do *Dataset* atribuindo 70% do *dataset* para treino (cerca de 3 milhões) e 30% para teste (cerca de 1.5 milhões).

O passo seguinte foi a construção de um Variational Autoencoder (VAE). Este tipo de modelo é capaz de aprender uma representação latente dos dados, permitindo uma compreensão eficiente das imagens (A. Roy, 2020). É constituído pelo codificador (Encoder), decodificador (Decoder) e a rede VAE em si.

O *Encoder* utiliza camadas convolucionais para extrair as características dos dados de entrada e, neste caso, um tamanho de kernel de apenas 1 pois as imagens são a preto e branco.

```
1.
2.  class Encoder(nn.Module):
3.      def __init__(self, latent_dim=16):
4.          super(Encoder, self).__init__()
5.          self.conv1 = nn.Conv2d(1, 32, 3, stride=2, padding=1)
6.          self.conv2 = nn.Conv2d(32, 64, 3, stride=2, padding=1)
7.          self.conv3 = nn.Conv2d(64, 128, 3, stride=2, padding=1)
8.          self.conv4 = nn.Conv2d(128, 256, 3, stride=2, padding=1)
9.          self.fc_mu = nn.Linear(256*8*8, latent_dim)
10.         self.fc_logvar = nn.Linear(256*8*8, latent_dim)
11.
12.     def forward(self, x):
13.         x = nn.functional.relu(self.conv1(x))
14.         x = nn.functional.relu(self.conv2(x))
15.         x = nn.functional.relu(self.conv3(x))
16.         x = nn.functional.relu(self.conv4(x))
17.         x = x.view(x.size(0), -1)
18.         mu = self.fc_mu(x)
19.         logvar = self.fc_logvar(x)
20.         return mu, logvar
```

Listagem 16: Encoder do VAE

As camadas convolucionais (conv1, conv2, conv3, conv4) são responsáveis por aprender as representações hierárquicas dos dados, reduzindo a dimensionalidade à medida que a informação é propagada pela rede. Em seguida, as saídas das camadas convolucionais são achatadas e passadas por duas camadas totalmente conectadas, que geram os parâmetros da distribuição latente (média e log-variância).

O decodificador é responsável por reconstruir os dados de entrada a partir da representação latente. O *Decoder* utiliza camadas totalmente conectadas para mapear as amostras latentes

de volta para o espaço de alta dimensionalidade dos dados originais. As amostras latentes são passadas por uma camada totalmente conectada (fc) que expande a dimensionalidade dos dados.

```
1.  class Decoder(nn.Module):
2.      def __init__(self, latent_dim=16):
3.          super(Decoder, self).__init__()
4.          self.fc = nn.Linear(latent_dim, 256*8*8)
5.          self.conv1 = nn.ConvTranspose2d(256, 128, 4, stride=2, padding=1)
6.          self.conv2 = nn.ConvTranspose2d(128, 64, 4, stride=2, padding=1)
7.          self.conv3 = nn.ConvTranspose2d(64, 32, 4, stride=2, padding=1)
8.          self.conv4 = nn.ConvTranspose2d(32, 1, 4, stride=2, padding=1)
9.
10.     def forward(self, z):
11.         x = self.fc(z)
12.         x = x.view(x.size(0), 256, 8, 8)
13.         x = nn.functional.relu(self.conv1(x))
14.         x = nn.functional.relu(self.conv2(x))
15.         x = nn.functional.relu(self.conv3(x))
16.         x = torch.sigmoid(self.conv4(x))
17.         return x
18.
```

Listagem 17: Decoder do VAE

Em seguida, a saída da camada totalmente conectada é remodelada para ter as dimensões adequadas para as operações de convolução reversa. As camadas convolucionais (conv1, conv2, conv3, conv4) são aplicadas ao longo do decodificador, aumentando gradualmente a dimensionalidade dos dados até que a reconstrução final seja obtida. A função de ativação ReLU é aplicada entre as camadas convolucionais para introduzir não-linearidade, e a função de ativação sigmoid é aplicada à última camada convolucional para obter a reconstrução final entre os valores 0 e 1.

A classe VAE combina o *Encoder* e o *Decoder*, além de definir o método “reparameterize” para amostrar as latentes de acordo com a distribuição aprendida pelo codificador. O VAE recebe os dados de entrada e realiza a passagem pelos componentes do codificador e decodificador. Retorna a representação latente da imagem e reconstrução final, para além da métrica de erro aplicada. Neste caso decidiu-se utilizar a MSE que calcula a média dos erros ao quadrado.

```
1.
2.  class VAE(nn.Module):
3.      def __init__(self, latent_dim=16):
4.          super(VAE, self).__init__()
5.          self.latent_dim = latent_dim
6.          self.encoder = Encoder(latent_dim)
7.          self.decoder = Decoder(latent_dim)
8.
9.      def reparameterize(self, mu, logvar):
```

```
        std = torch.exp(0.5 * logvar)
        eps = torch.randn_like(std)
        return eps * std + mu

    def forward(self, x):
        mu, logvar = self.encoder(x)
        z = self.reparameterize(mu, logvar)
        x_hat = self.decoder(z)
        return x_hat, mu, logvar
```

Listagem 18: VAE

Foram definidos dois modelos VAE, um para 128x128 e outro para 256x256. A configuração da rede para 256x256 pode ser encontrada em anexo.

4.1.6.3 Treino do Modelo

Com a rede configurada, foram definidas as funções de treino e de teste. A primeira tinha como objetivo alimentar o modelo com as imagens, calcular a taxa de erro e otimizar os parâmetros do modelo de modo a melhorar a sua performance. A última, para teste num set de dados nunca visto (*set* de validação), de modo a avaliar o desempenho do modelo nessas condições. As funções descritas podem ser encontradas em anexo.

Os modelos foram treinados para 200 épocas, foram passadas 256 imagens por vez ao modelo (batch_size), o tamanho do vetor latente foi definido para 256 elementos e a taxa de aprendizagem (learning_rate) foi inicialmente definida para $1e^{-3}$.

```

diz_loss = {'train_loss': [], 'val_loss': []}
best_valid_loss = 100000
counter = 0
new_lr = lr
with mlflow.start_run() as run:
    for epoch in range(epochs):
        train_loss = train_epoch(
            model, device, train_loader, loss_function, optimizer)
        val_loss = test_epoch(model, device, test_loader, loss_function)
        print('\n EPOCH {}/{} \t train loss {} \t val loss {}'.format(epoch +
              1, epochs, train_loss, val_loss))
        if val_loss.item() < best_valid_loss:
            counter = 0
            best_valid_loss = val_loss.item()
            print(f"\nBest validation loss: {best_valid_loss}")
            print(f"\nSaving best model for epoch: {epoch+1}\n")
            torch.save({
                'epoch': epoch+1,
                'model_state_dict': model.encoder.state_dict(),
                'optimizer_state_dict': optimizer.state_dict(),
                'loss': criterion,
            }, '/code/data/models/best_encoder_vae_128.pth')
```

```
            torch.save({
                'epoch': epoch+1,
                'model_state_dict': model.decoder.state_dict(),
                'optimizer_state_dict': optimizer.state_dict(),
                'loss': criterion,
            }, '/code/data/models/best_decoder_vae_128.pth')
            torch.save({
                'epoch': epoch+1,
                'model_state_dict': model.state_dict(),
                'optimizer_state_dict': optimizer.state_dict(),
                'loss': criterion,
            }, '/code/data/models/best_model_vae_128.pth')
        else:
            counter += 1

        if counter == 10:
            print(f"\nPrevious LR {new_lr} \t New Learning Rate {new_lr/2}")
            counter = 0
            new_lr = new_lr/2
            optimizer = Adam(model.parameters(), lr=new_lr)

        mlflow.log_metric("MSE", val_loss.item())
        mlflow.log_metric("epoch", epoch+1)

        diz_loss['train_loss'].append(train_loss)
        diz_loss['val_loss'].append(val_loss)
        plot_ae_outputs(model, n=20)
```

Listagem 19: Função que efetua o treino e teste do modelo e guarda o melhor

Foi efetuada uma técnica de treino em que a taxa de aprendizagem era reduzida para metade a cada 10 épocas, se não houvesse melhoria do mesmo (*learning rate scheduling*). Isto permitiu que o modelo se tornasse mais estável e consistente no treino, não tomando decisões tão diferentes. As métricas foram guardadas no MlFlow, que é uma plataforma com a finalidade de permitir uma abordagem mais sistemática e colaborativa para o desenvolvimento dos modelos. À medida que o modelo ia melhorando, as suas partes eram guardadas para uso futuro.

Figura 49: Treino do Modelo

4.1.6.4 Comparação do Modelo

Com o objetivo da avaliação do modelo e identificação de melhores abordagens, tornou-se necessário a comparação dos modelos criados, um com o outro (128 e 256) e com outras abordagens.

Primeiramente, comparando os dois modelos implementados, foi averiguado que a principal diferença nos valores da métrica de erro MSE, vinha principalmente pela definição da métrica. Por outras palavras, esta avalia o número de pixéis certos entre a imagem original e a falsa, as imagens são compostas maioritariamente por pixéis pretos, portanto caso o modelo de 256 por 256 acerte todos os pretos, já vai ter vantagem sobre o outro que tem menos pixéis pretos. Deste modo, devido aos valores das métricas não serem assim tão diferentes na ordem de grandeza (0.01002 para 128x128 e 0.00677 para 256x256, ambos os valores para uma *sample* do *dataset*) e do treino do modelo ser bastante mais rápido para os valores de 128x128, este foi preferido em relação ao outro.

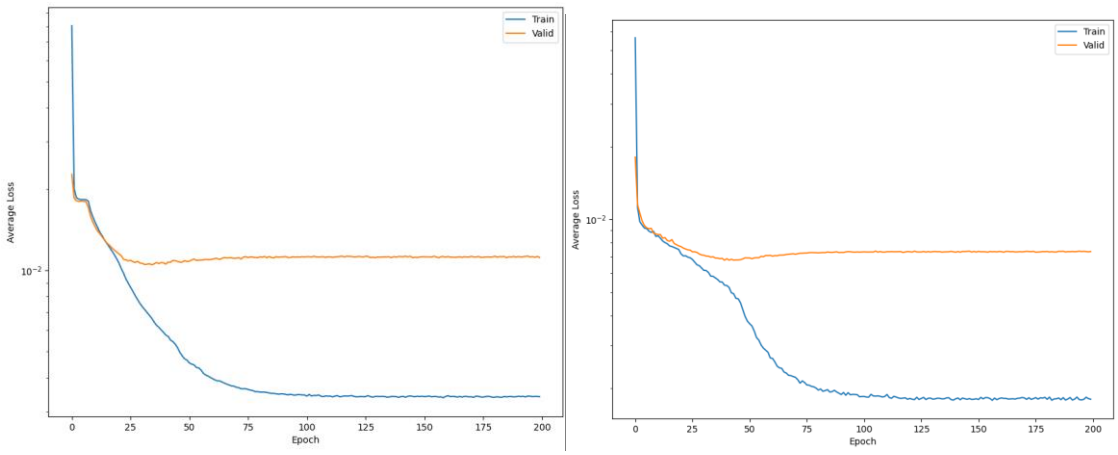


Figura 50: esquerda: Gráfico de Comparação da Métrica de Erro de Treino e Validação do Modelo 128x128 para uma *Sample* do Dataset direita: Gráfico de Comparação da Métrica de Erro de Treino e Validação do Modelo 256x256 para uma *Sample* do Dataset

De seguida, procurou-se a comparação com diferentes arquiteturas. O supervisor externo, Luís Costa, já tinha construído um *autoencoder*, então tornou-se inevitável a comparação. Para além deste, foi encontrado um repositório no *github* (GitHub, n.d.) com diversas arquiteturas de *autoencoders* e *GANs* e foram escolhidas duas arquiteturas para serem usadas como comparação. Foram escolhidas as arquiteturas que, segundo o autor, apresentavam os melhores valores de MSE (RAE-L2) e de FID (VAEGAN) (Chadebec et al., 2022).

O RAE-L2 (Regularized Autoencoder with L2-norm) é uma extensão do autoencoder tradicional, que visa melhorar a capacidade de construção das imagens. Adiciona uma regularização L2, também conhecida como penalidade de norma L2, ao processo de reconstrução. Esta regularização ajuda a evitar o *overfitting*, reduzindo a complexidade do modelo e favorecendo a generalização dos dados (Chadebec et al., 2022). O VAEGAN (Variational Autoencoder Generative Adversarial Network) é uma combinação de duas técnicas, o VAE e a GAN. Aproveita a capacidade do VAE de aprender representações latentes ricas e da GAN de gerar amostras sintéticas realistas. Isto permite uma geração mais avançada e diversificada de dados (Chadebec et al., 2022).

O *autoencoder* do Luís não precisou de alterações para funcionar com o *dataset* deste projeto, porém os restantes tiveram de ser alterados pois estavam preparados para imagens RGB, ou seja 3 *kernels*, de 28 por 28 pixéis, pois estes usavam como exemplo o *dataset* MNIST. De realçar que o *dataset* MNIST é semelhante ao deste projeto.

A função de *loop* das épocas foi a mesma utilizada para o modelo construído para este projeto, sendo que a função de treino foi adaptada para alimentação dos modelos em questão. Os gráficos de treino dos modelos podem ser encontrados em anexo.

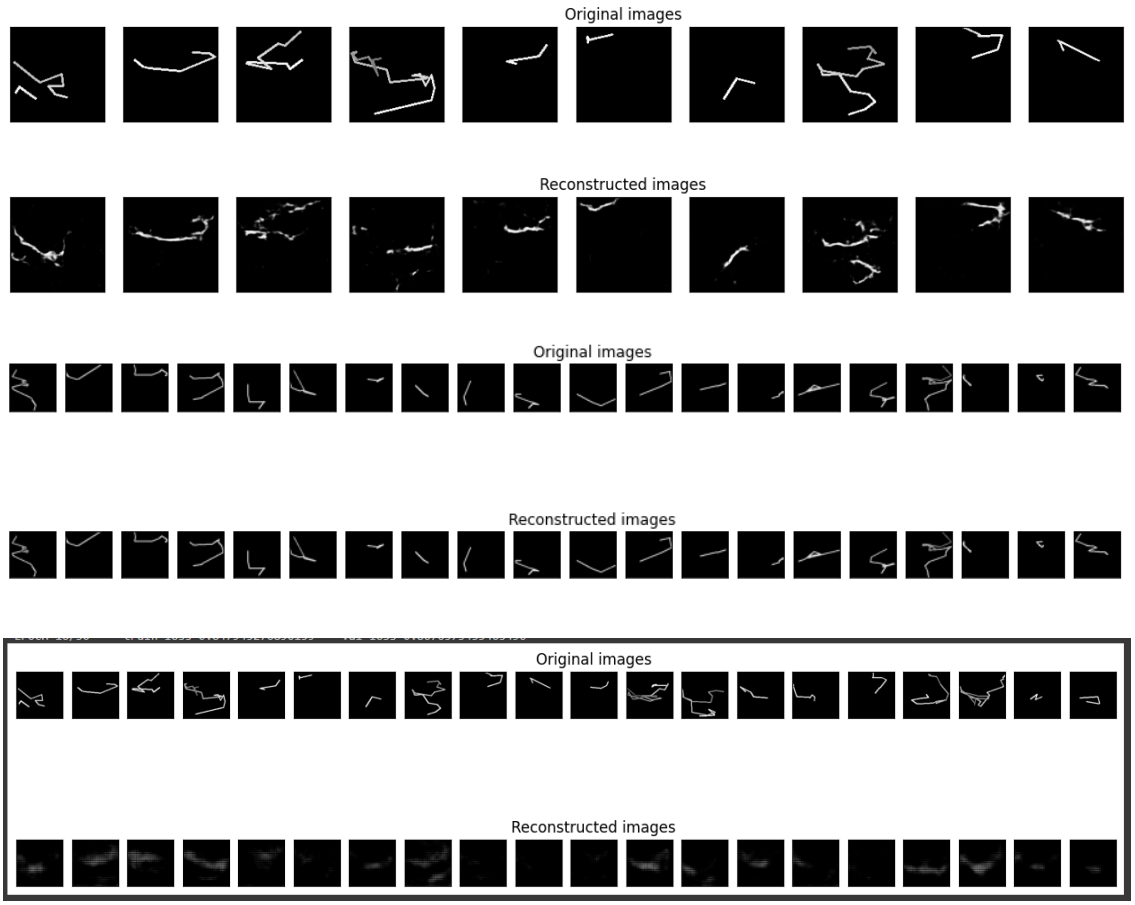


Figura 51: Exemplo do Treino dos Modelos, de cima para baixo: Autoencoder do Luís, RAE-L2 e VAEGAN

Comparando os diferentes modelos com o escolhido anteriormente (128x128), foi escolhido o modelo construído (VAE), pois este apresentou os melhores resultados.

Tabela 7: Comparação dos Resultados dos Modelos

Modelo	Resultado (MSE)
VAE	0.002592
Autoencoder Luís	0.008107
RAE-L2	0.003497
VAEGAN	0.011245

O *encoder* do modelo escolhido foi então usado para o passo seguinte.

4.1.6.5 Obtenção do Vetor Latente e Redução de Dimensionalidade

Após selecionar o melhor modelo, realizou-se a obtenção do vetor latente através do *encoder*. O retorno do *encoder* era um vetor de 256 elementos. Isto quer dizer que uma imagem de uma sequência estava representada nesses 256 elementos.

```
1.
2.  full_teams_sequences_encoding_df = pd.DataFrame([])
3.
4.  for seq_path in all_sequences_paths:
5.      if seq_path.endswith('.db'):
6.          continue
7.
8.      team_id = seq_path.split('/')[8]
9.      competition_id = seq_path.split('/')[7]
10.     season_id = seq_path.split('/')[6]
11.     sequence_id = seq_path.split('/')[-1].split('.')[0]
12.     match_id = sequence_id.split('_')[0]
13.     encoded_samples = []
14.     img = Image.open(seq_path)
15.     img_tensor = test_transform(img).unsqueeze(0).to(device)
16.     with torch.no_grad():
17.         mu, log_var = encoder(img_tensor)
18.     encoded_img = reparameterize(mu, log_var)
19.     encoded_img = encoded_img.flatten().cpu().numpy()
20.     encoded_sample = {f"Enc. Variable {i}": enc for i,
21.         enc in enumerate(encoded_img)}
22.
23.     encoded_samples.append(encoded_sample)
24.     encoded_samples_df = pd.DataFrame(encoded_samples)
25.     encoded_samples_df['season_id'] = season_id
26.     encoded_samples_df['competition_id'] = competition_id
27.     encoded_samples_df['team_id'] = team_id
28.     encoded_samples_df['sequence_id'] = sequence_id
29.     encoded_samples_df['match_id'] = match_id
30.     full_teams_sequences_encoding_df =
    pd.concat([full_teams_sequences_encoding_df, encoded_samples_df])
```

Listagem 20: Código da Obtenção do Vetor Latente

Estes 256 elementos poderiam ser vistos como 256 dimensões, porém caso estas 256 variáveis pudessem ser reduzidas para 2, seria possível apresentar os valores como pontos num gráfico bidimensional, tornando possível a identificação de agrupamentos, padrões e correlações entre os dados.

Com este objetivo em mente, utilizou-se a técnica t-SNE (t-Distributed Stochastic Neighbor Embedding) para reduzir a dimensionalidade de 256 variáveis para apenas 2. O t-SNE é um algoritmo de redução de dimensionalidade amplamente utilizado para visualização de dados complexos. Este mapeia os dados de alta dimensionalidade para um espaço de menor dimensionalidade, preservando as relações e estruturas entre os pontos (Violante, 2018). Com essa redução, conseguiu-se simplificar a visualização e interpretação dos dados. Em anexo está apresentada uma representação de todas as sequências de acordo com a representação do t-SNE e o agrupamento pelas equipas.

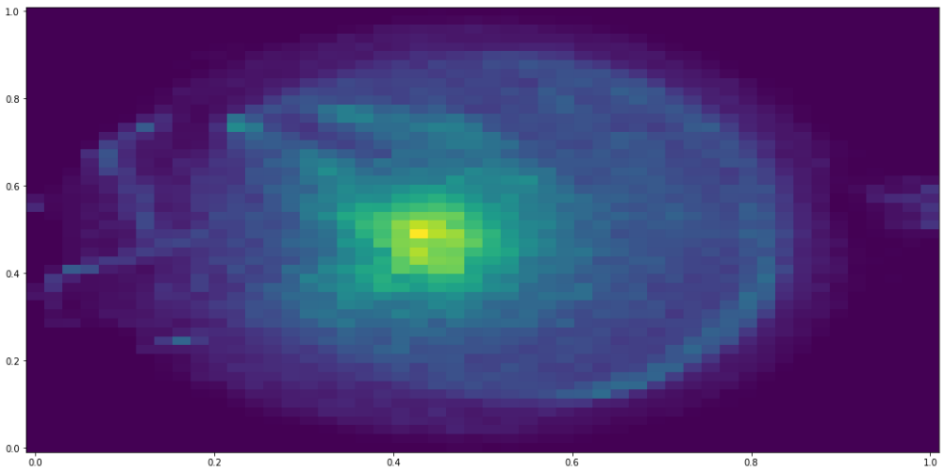


Figura 52: Resultados do t-SNE

A grelha de representação destes resultados era de 50 por 50. Tornou-se também necessário o cálculo dos intitulados *buckets*, que eram os pixéis da grelha, para representação na aplicação web.

```
1.
2.  tsne_dataset['bucket_id'] = None
3.
4.  for i, row in tsne_dataset.iterrows():
5.      coordinate = (row['tx'], row['ty'])
6.      for bucket_coord, bucket_id in bucket_ids.items():
7.          x_interval = bucket_coord[0]
8.          y_interval = bucket_coord[1]
9.
10.         if x_interval[0] <= coordinate[0] <= x_interval[1] and y_interval[0] <=
    coordinate[1] <= y_interval[1]:
11.             tsne_dataset.at[i, 'bucket_id'] = bucket_id
12.             break
```

Listagem 21: Código do cálculo do Bucket

Os dados dos resultados do t-SNE com o respetivo *bucket* foram guardados numa tabela do SQL Server.

4.1.6.6 Cálculo das Equipas Semelhantes

Para o cálculo das equipas semelhantes foram também utilizados os vetores latentes resultantes do Encoder. Filtrando os valores para a granularidade da equipa, foi feita a média sobre cada um dos 256 elementos dos vetores latentes. Além disso, também decidiu-se incluir os valores dos indicadores estatísticos previamente calculados para as equipas, neste cálculo.

Para este cálculo foram utilizadas apenas sequências que terminaram em remate devido a uma limitação de hardware.

```
1.
2.  from sklearn.preprocessing import StandardScaler
3.
4.  scaler = StandardScaler()
5.  cos_sim_matrix =
    cosine_similarity(scaler.fit_transform(team_sequences_stats.drop(columns=['team_id
    ']).values.tolist()))
6.
7.  team_ids = team_sequences_stats['team_id'].tolist()
8.  similar_teams = []
9.  for i in range(len(team_ids)):
10.     for j in range(i + 1, len(team_ids)):
11.         team1 = team_ids[i]
12.         team2 = team_ids[j]
13.         similarity = cos_sim_matrix[i, j]
14.         similar_teams.append((team1, team2, similarity))
15. similar_teams.sort(key=lambda x: x[2], reverse=True)
```

Listagem 22: Similaridade de Cosseno

Assim, para determinar a similaridade foi utilizada a técnica de similaridade de cosseno (cosine similarity) para as diferentes equipas com os valores codificados das sequências e os valores estatísticos. Esta técnica permite comparar a direção e magnitude dos vetores, avaliando o quão semelhantes são, neste caso, as equipas em termos de estilo de jogo e desempenho estatístico (Han et al., 2012).

A avaliação do resultado e comparação com outros serviços que apresentam a comparação entre equipas está presente no capítulo 4.2.

4.1.7 Construção da Aplicação Web

Uma aplicação web foi desenvolvida para visualização dos resultados do projeto. Neste caso, foi desenvolvido um sólido *backend* utilizando a linguagem de programação c# e a *framework* Mediator. Esta escolha permitiu uma estrutura organizada e modularizada para lidar com as operações e a lógica de negócios por trás da aplicação. O Mediator facilitou a comunicação entre os diferentes componentes do *backend*, tornando o código mais legível e escalável.

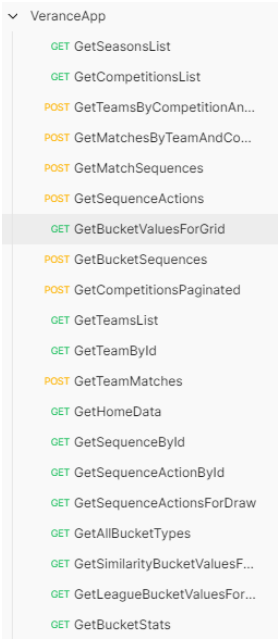


Figura 53: Lista de Todos os Endpoints implementados

É de realçar ainda que o *backend* seguiu a estrutura da ”Clean Architecture”, conforme analisado e projetado no Capítulo 3 deste documento. Esta abordagem arquitetural garantiu uma separação de responsabilidades e uma organização clara do código, promovendo modularidade, testabilidade e manutenibilidade.

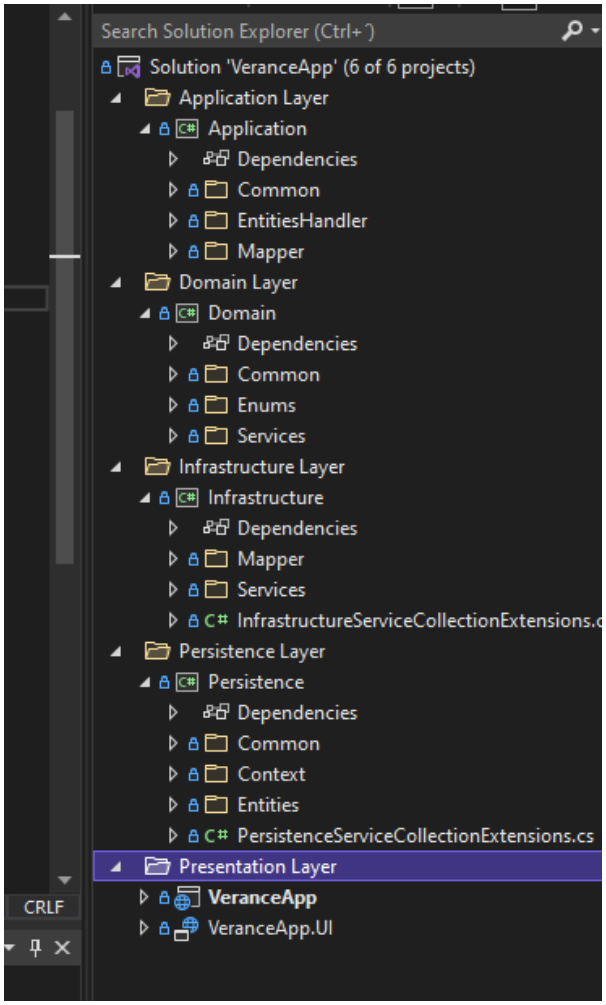


Figura 54: Arquitetura Backend

Para a integração entre o *backend* e o *frontend*, foi utilizado o NSwag, uma ferramenta que permitiu gerar automaticamente os métodos de conexão entre as duas partes. Com o NSwag, foi possível definir os *endpoints* da API no *backend* e criar automaticamente os serviços correspondentes no *frontend*. Isso agilizou o processo de desenvolvimento e reduziu a probabilidade de erros de integração.

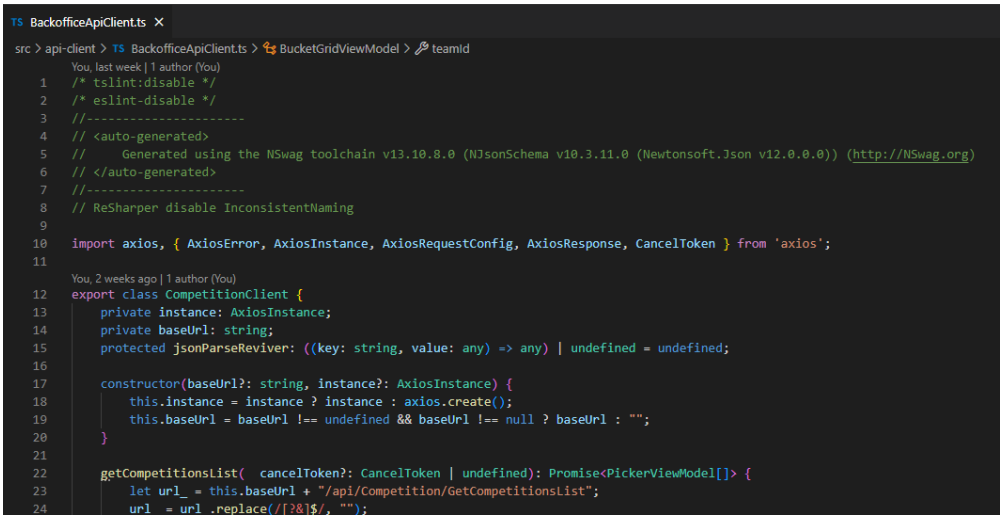


Figura 55: Ficheiro Criado pelo NSwag com a integração do Backend e Frontend

O *frontend* foi construído em TypeScript e React, usando o template Vuexy (PIXINVENT, n.d.) de base.

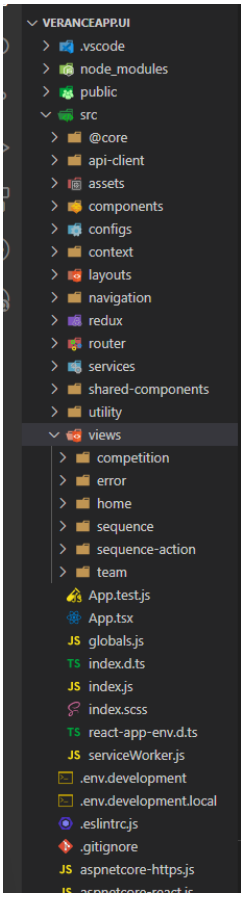


Figura 56: Arquitetura Frontend

O TypeScript proporcionou uma camada adicional de segurança e tipagem estática, garantindo uma melhor manutenção do código e prevenindo erros comuns. O React, por sua vez, ofereceu uma abordagem eficiente para criar interfaces mais interativas e responsivas.

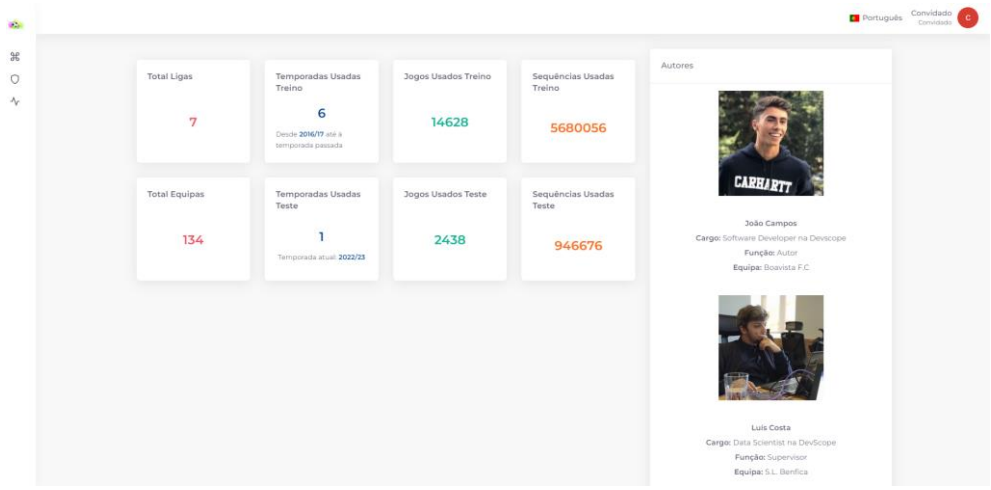


Figura 57: Menu Inicial da Aplicação

Esta aplicação oferece várias funcionalidades atendendo aos requisitos identificados e os casos de uso analisados. Começando com a listagem das competições disponíveis, os utilizadores podem explorar e selecionar uma liga especifica para serem listadas todas as equipas referentes à liga escolhida.

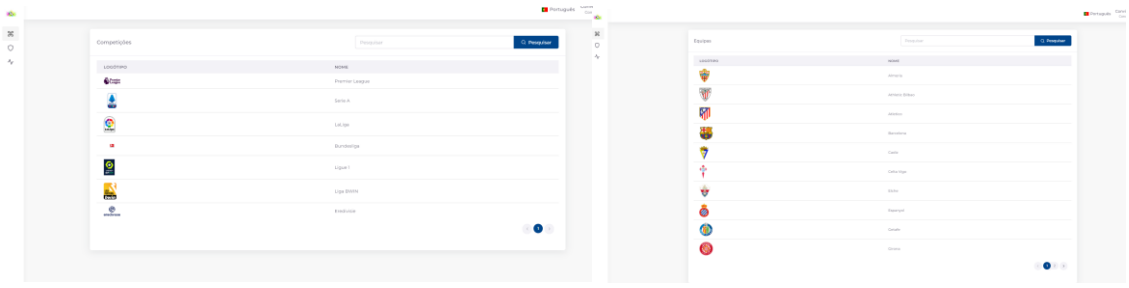


Figura 58: Listagens das Competições e Equipas

Além disso, existem recursos de filtragem que permite aos utilizadores visualizar e navegar mais facilmente entre as equipas.

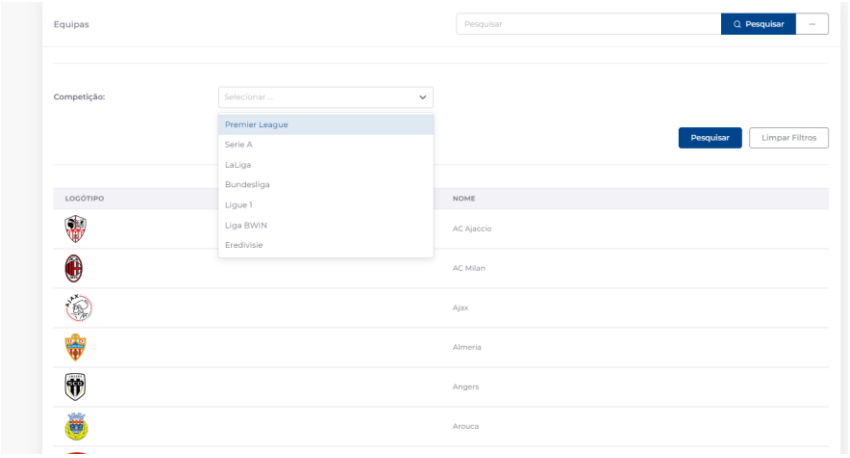


Figura 59: Filtro de Competições na Listagem de Equipas

Ao acessar a página de detalhes de uma equipa, os utilizadores têm acesso a variadas estatísticas calculadas ao longo do projeto, conforme especificado no subcapítulo 4.1.5, listagem de todos os jogos da equipa na atual temporada e apresentação das equipas semelhantes.

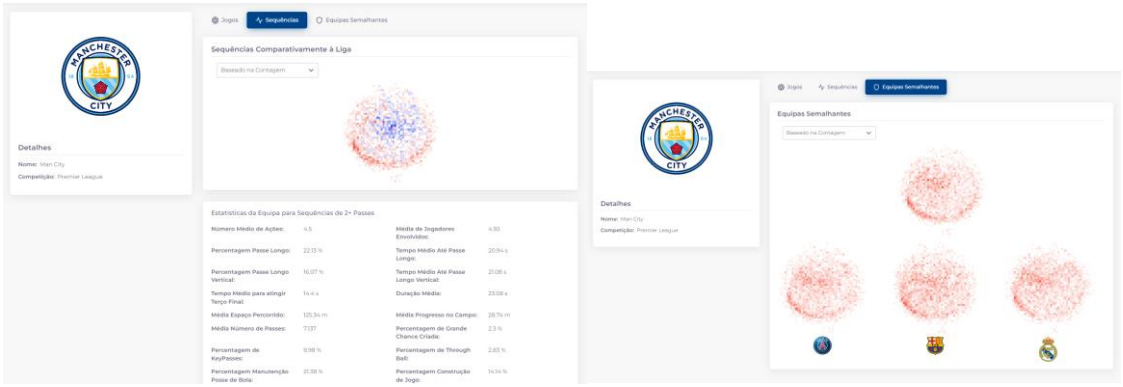


Figura 60: Página de detalhes da Equipa com as Estatísticas e Equipas Semelhantes para o Manchester City

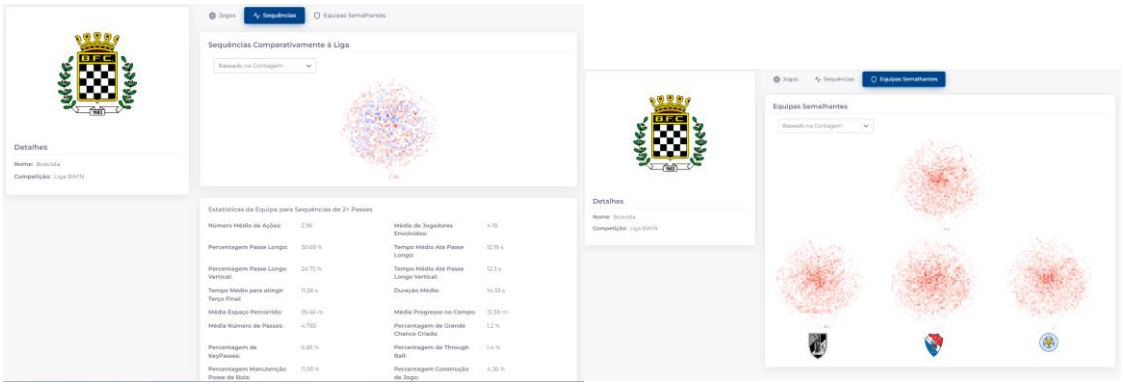


Figura 61: Página de detalhes da Equipa com as Estatísticas e Equipas Semelhantes para o Boavista

Para além disto, a listagem dos jogos reencaminha para a listagem das sequências relativas ao jogo selecionado.

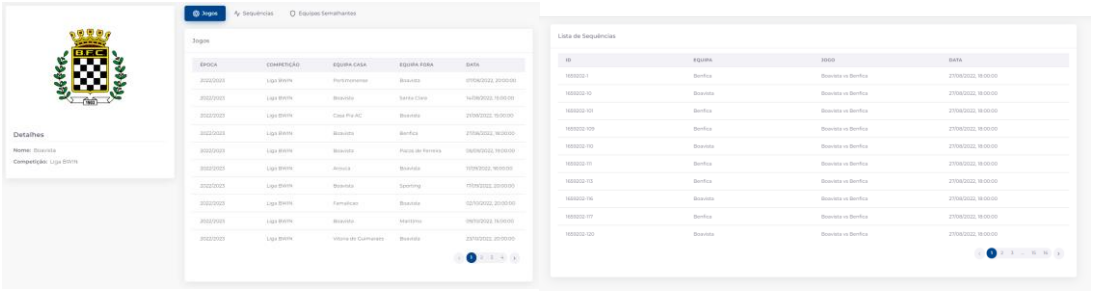


Figura 62: Listagem de Jogos e Sequências do Jogo Selecionado

Ao acessar a página de detalhes de uma sequência, os utilizadores encontrarão estatísticas detalhadas, como tempo de posse de bola, passes completados, entre outras informações relevantes. Além disso, a sequência será apresentada visualmente num desenho, proporcionando uma experiência imersiva.

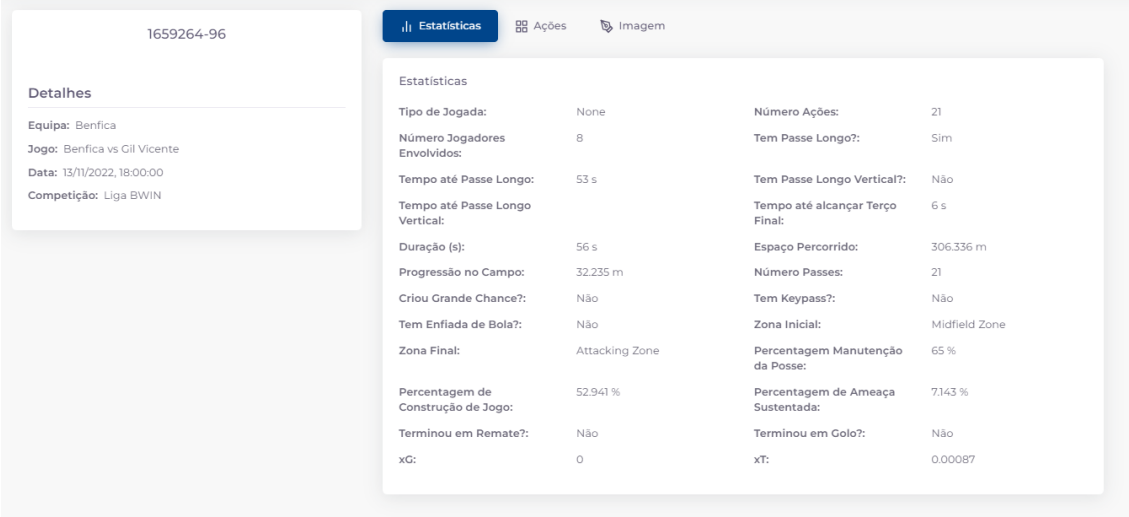


Figura 63: Secção de Estatísticas da Página de Detalhe da Sequência

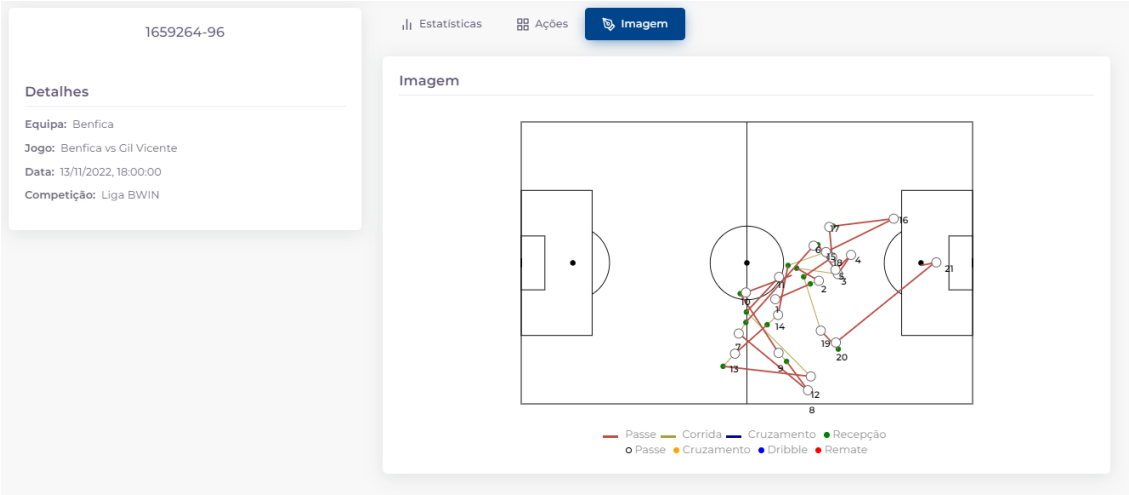


Figura 64: Secção do Desenho da Sequência na Página de Detalhe da Sequência

Por fim, passando ao nível mais profundo de granularidade, na página de detalhes da sequência está presenta uma listagem de todas as ações da sequência que redireciona para a página de detalhes da ação, apresentando todas as estatísticas da mesma.

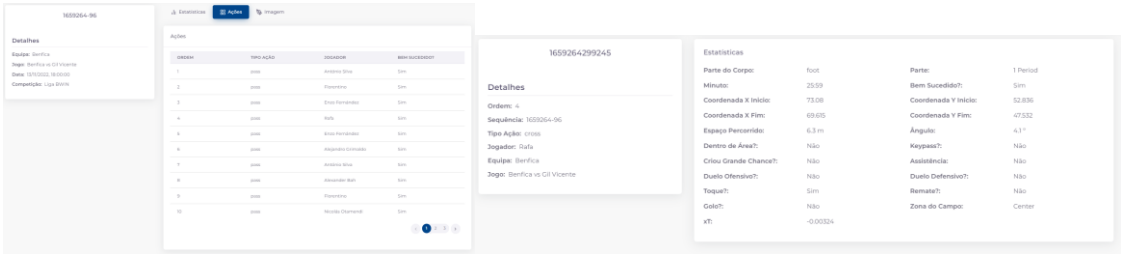


Figura 65: Listagem de Ações e Página de Detalhes da Ação

Para além de tudo isto, criou-se uma página das sequências, acessada através da navegação lateral, que apresenta uma visão geral de todas as sequências da temporada e permite a visualização das mesmas pertencentes a determinado bucket.

Figura 66: Página Principal das Sequências

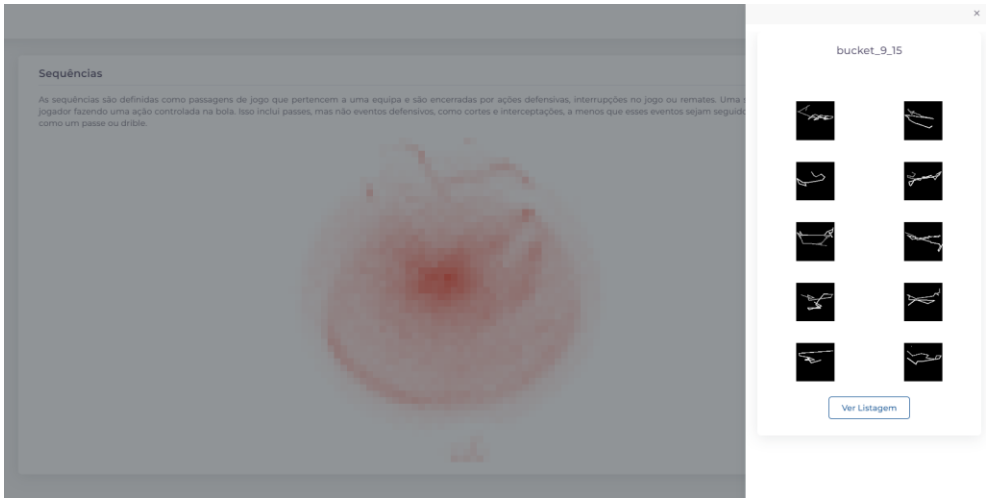


Figura 67: Secção Lateral que apresenta a visualização das Sequências depois de selecionado um *bucket*

A secção lateral que possui a visualização das sequências, reencaminha para a listagem das sequências pertencentes ao bucket selecionado e esta listagem possui a funcionalidade de filtragem pela equipa.

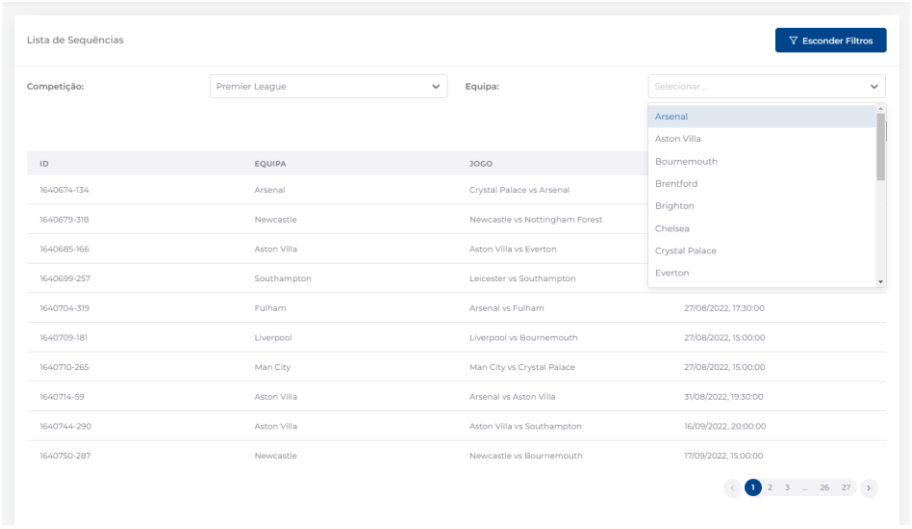


Figura 68: Filtragem por Equipa na Listagem das Sequências

Esta funcionalidade extra permitiu o acesso a uma sequência especifica de forma mais rápida e eficaz.

Em resumo, a construção desta aplicação web apresenta todos os resultados obtidos no projeto e oferece aos utilizadores uma plataforma abrangente para explorar informações de ligas, equipas, sequências de jogadas e ações individuais. Com recursos a filtragem, estatísticas detalhadas, visualizações gráficas e comparação entre equipas, os utilizadores poderão se envolver com o mundo do futebol de forma interativa e informativa.

4.2 Avaliação da solução

Sendo esta solução composta por vários componentes individuais, a sua avaliação deve ser feita avaliando cada componente individualmente. Os componentes dignos de avaliação são: o modelo de xG, o modelo de xT e a componente de aprendizagem não supervisionada.

4.2.1 Avaliação do modelo de xG

Neste contexto, várias métricas foram utilizadas, como precision, recall e F1-score, para avaliar o desempenho do modelo em relação à classificação correta dos golos. Foi utilizado um *threshold* de 0.5 para o resultado do modelo de xG para determinar se se tratava de um golo para se puderem calcular as métricas mencionadas.

A métrica de precision mede a proporção de golos previstos corretamente em relação ao total de golos previstos pelo modelo. Quanto maior a precision, maior a confiabilidade das previsões de golo.

A métrica de recall, por sua vez, avalia a proporção de golos corretamente previstos em relação ao total de golos reais. Mede a capacidade de o modelo identificar corretamente os golos que realmente ocorreram. Um valor alto de recall indica que o modelo é eficaz em identificar a ocorrência real de golos.

O F1-score é uma medida combinada de precision e recall. Fornece uma medida única do desempenho do modelo, considerando tanto a precisão quanto a capacidade de identificar corretamente os golos.

Os resultados destas métricas para os modelos foram os seguintes:

Tabela 8: Métricas de Avaliação do Modelo de xG

Métrica	Valor
Precision	0.7869
Recall	0.5105
F1-Score	0.6193

A partir da definição das métricas e observando os valores obtidos é possível afirmar que este modelo prevê corretamente a maioria das vezes, mas não consegue identificar corretamente todos os exemplos positivos.

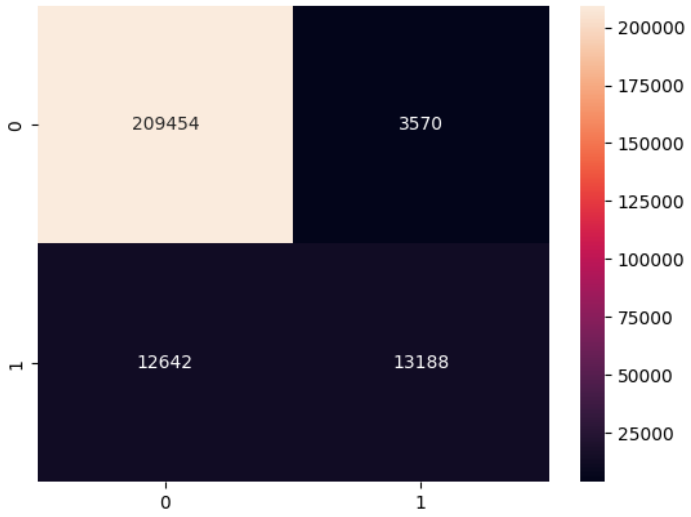


Figura 69: Matriz de Confusão

Além das métricas, foi construída uma matriz de confusão. Esta matriz mostra a quantidade de golos corretamente previstos (verdadeiros positivos), os golos previstos incorretamente (falsos positivos), os golos não previstos (falsos negativos) e os golos corretamente não previstos (verdadeiros negativos).

Outro método de avaliação utilizado foi a verificação dos resultados de alguns jogos com os valores de xG apresentados pelo modelo e a comparação com outros serviços. Neste caso, os serviços utilizados como meio de comparação foram o Flashscore e o FBRef que utiliza métricas da Statsbomb.

Tabela 9: Comparação Resultados do Modelo com Resultados Reais e outros Serviços

Jogo	Resultado Real	Resultado Esperado	Resultado Flashscore	Resultado FBRef
Benfica vs Santa Clara	3 – 0	2.87 – 1.29	2.41 – 1.07	2.4 – 1.0
Colónia vs Bayern	1 – 2	1.10 – 1.67	1.78 – 0.91	1.3 – 1.3
Valladolid vs Getafe	0 – 0	0.71 – 0.34	0.86 – 0.42	0.9 – 0.5
Man City vs Arsenal	4 - 2	2.42 – 0.47	2.22 – 0.52	2.5 – 0.5
Atalanta vs Monza	5 - 2	4.49 – 1.58	3.60 – 0.87	4.0 – 1.2

Comparativamente aos valores apresentados pelos serviços comparados, os valores calculados aproximam-se bastante um dos outros, no entanto é possível verificar que os valores obtidos

pelo modelo construído são, relativamente melhores, na medida em que se aproximam mais dos valores reais.

4.2.2 Avaliação do modelo de xT

O modelo de xT não pôde ser adequadamente avaliado, pois a sua implementação foi baseada exclusivamente na aplicação direta da definição do cálculo estabelecido pelo criador da métrica, Karun Singh.

Infelizmente, não foram encontrados serviços ou referências comparáveis disponíveis que pudessem ser utilizados para uma avaliação mais abrangente do modelo. Dessa forma, a avaliação restringiu-se à validação da fórmula e à verificação da consistência interna dos resultados obtidos. A ausência de um padrão estabelecido e de bases de dados confiáveis para comparação limitou a capacidade de avaliar a eficácia e a precisão do modelo de xT em relação a outros modelos com o mesmo objetivo.

4.2.3 Avaliação do Autoencoder

O módulo de visão não supervisionada, especificamente o autoencoder implementado, foi avaliado utilizando diferentes métricas para avaliar a qualidade dos resultados obtidos. Uma métrica amplamente utilizada nesse contexto é o Mean Squared Error (MSE), que permite medir a diferença entre as imagens originais e as imagens reconstruídas pelo autoencoder. O MSE calcula a média dos quadrados das diferenças pixel a pixel entre as duas imagens, fornecendo uma medida quantitativa de quão bem o modelo é capaz de reconstruir as entradas. O valor da MSE para o melhor modelo foi de 0.002592.

A avaliação do resultado final, especificamente do cálculo das equipas semelhantes, enfrentou um desafio particular. Infelizmente, não foi possível encontrar serviços gratuitos que fornecessem os resultados desejados para comparação entre as equipas. Diante dessa limitação, o método de avaliação adotado foi baseado no conhecimento e na expertise do autor e do supervisor do projeto. Primeiramente, foram utilizados os resultados da técnica do t-SNE, de seguida, os valores do vetor latente do *encoder* e, por último, a versão escolhida, os dados do vetor latente em conjunto das estatísticas da equipa. Os resultados desta última versão atenderam às espectativas do autor e supervisor.

5 Conclusões

Este capítulo concentrar-se-á na exposição de uma visão geral da tese, identificação dos objetivos concretizados, na apresentação das limitações sentidas por parte do autor e de possíveis melhorias para trabalhos futuros. Por fim, é feita uma apreciação final sobre o projeto.

5.1 Visão Geral

Para melhor identificar o problema de pesquisa e esclarecer as motivações e objetivos subjacentes a este trabalho, foi realizada uma revisão bibliográfica abrangendo vários temas, quer no contexto futebolístico, quer mais abrangentes.

Foram apresentados os tipos de dados utilizados no futebol, incluindo dados de fluxo de eventos (informações coletadas e registadas durante um jogo) e dados de rastreamento ótico (informações registadas sobre os movimentos dos jogadores em campo). Foram também mencionadas as empresas que fornecem esses dados, como Opta e Prozone, e os desafios que surgem da diversidade de *providers* de dados, como falta de informação, terminologia e definições diferentes e a dificuldade de aplicar ferramentas de análise automática.

Após isto, foi apresentado o conceito da análise avançada no futebol, que envolve o uso de técnicas complexas de análise de dados, como aprendizagem automática e visualização de dados, para extrair informações valiosas e melhorar o desempenho. Os conceitos de posse e sequência, que são centrais para análises avançadas, são descritos. A posse de bola é definida como uma ou mais sequências seguidas pertencentes à mesma equipa, enquanto uma

sequência é uma passagem de jogo que pertence a uma equipa e é finalizada por ações defensivas, interrupções no jogo ou remates. Através da análise das sequências é possível perceber padrões e classificá-las de acordo com esses padrões. Os tipos de sequências explicados são a manutenção da posse de bola, construção do jogo, ameaça contínua, ritmo rápido, jogo direto, contra-ataque, cruzamento e pressão alta.

Após isto, o foco passa para os conceitos relativos à IA. São descritos os conceitos de aprendizagem automática (ML) e aprendizagem profunda (DL), que são subcampos da inteligência artificial que envolvem o projeto. E é também dado especial foco e são descritos os modelos mais utilizados nesta área da análise desportiva, como as RNA, Autoencoders e GANs. É explicado que existem vários tipos de ML, incluindo aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço. Quando se trata de futebol, a IA está a ser usada para melhorar muitos aspetos da indústria, como rastrear as posições dos jogadores durante as transmissões, automatizar o conteúdo do jogo e identificar futuras estrelas.

Para além disto, são descritas as soluções existentes para análise de futebol, incluindo InStat Scout, Opta, StatsBomb IQ e Wyscout. Estas plataformas fornecem uma variedade de ferramentas e recursos, como visualização de dados, análise de desempenho, análise de vídeo e gerenciamento da equipa para ajudar treinadores, analistas e outras partes interessadas a obter informações sobre o desempenho da equipa e do atleta e tomar decisões informadas. São amplamente utilizados por vários clubes de futebol ao redor do mundo e são conhecidos pelas suas métricas avançadas e análises baseadas em dados.

Esta pesquisa concentrou-se principalmente na questão de projetar uma solução inovadora em vários aspetos. O modelo de xG implementado iria verificar a influência e impacto das ações anteriores no melhoramento do cálculo da métrica. Os modelos atuais apenas têm em consideração a ação atual e as suas características. O *autoencoder* variacional implementado constitui, por si só, também uma melhoria relativamente aos modelos comparados no projeto e possibilitou estabelecer uma relação entre os padrões extraídos das sequências e o estilo de jogo de uma equipa de futebol.

Foi feita uma análise dos casos de uso a implementar na solução e desenhada a arquitetura a seguir para alcançar os resultados pretendidos.

Após isto, foi realizada a implementação. Os dados foram obtidos através de ferramentas de *scrapping* da página de resultados desportivos WhoScored e foram estabelecidas duas *pipelines* para tratamento dos mesmos. A primeira tinha como objetivo a obtenção dos ids dos jogos e datas dos mesmos, definindo assim todo o calendário da época desportiva das várias ligas desportivas. A segunda tinha o objetivo de efetivamente obter os dados para os jogos anteriormente obtidos, nas datas dos mesmos. O processo de ETL envolvia também o tratamento de valores nulos, mapeamento de todos os valores para o formato OPTA e o registo no SQL Server.

Com os dados prontos e guardados, foram estabelecidas 3 frentes do projeto, com 3 objetivos distintos. A primeira era a criação do modelo de xG, a segunda a criação do modelo de xT e a terceira a construção de um *autoencoder* para extração dos padrões das imagens das sequências.

Para alcançar estes três objetivos, estava em falta uma questão, a construção das sequências a partir dos dados dos eventos. Para isto, foi definida uma função de construção que identificava cada ação com o id de sequência correspondente e a sua ordem dentro da sequência. Agora estavam reunidos todos os ingredientes para avançar nas três frentes do projeto.

Para o primeiro objetivo, foram analisados os dados e construídas as *features* necessárias para o cálculo do xG, para todas as ações. De seguida, foram estabelecidas quais as ações que antecediam quais, de modo a ter uma correspondência entre a ação atual e as ações passadas, neste caso as 3 ações passadas. Depois disto, foi filtrado o conjunto de dados para que as ações atuais correspondessem apenas a remates e, com isto, fosse apenas calculado o xG para ações de remates. Neste ponto tínhamos um conjunto de dados que correspondiam a características (*features*) das ações de remate e das 3 ações passadas. Tudo isto foi passado ao modelo e os resultados obtidos, pela comparação com os modelos do mercado, permitiram deduzir que as ações anteriores impactam no cálculo do xG e estes resultados foram superiores aos modelos dos serviços comparados, aproximando-se dos valores dos resultados reais dos jogos.

A construção do modelo de xT, o segundo objetivo do projeto, seguiu a definição da métrica criada por Karun Singh. O campo foi dividido numa grelha de 105 por 68. Todo o conjunto de ações foi passado ao modelo para que este calculasse a probabilidade de o jogador marcar golo, decidir rematar ou movimentar a bola para todas as células da grelha anteriormente calculada.

Após isto o cálculo era feito subtraindo o xT da ação na posição final pela da posição inicial, verificando qual o perigo adjacente ao movimentar a bola da célula inicial para a final.

Por fim, para o terceiro e último objetivo, foi construído um *autoencoder* que conseguisse extrair características e padrões das imagens das sequências. Primeiramente, foram construídas as imagens das sequências. Através do auxílio da ferramenta OpenCV estas foram desenhadas, utilizando as coordenadas das ações. Depois disto, foi construído e treinado um *autoencoder* variacional com estas mesmas imagens. Quanto melhor o modelo conseguisse reconstruir as imagens, significaria que melhor este conseguia representá-las através do vetor latente, neste caso de 256 elementos. O *encoder* treinado foi então utilizado para converter a representação dos dados da temporada atual para a representação latente. Foi utilizada a técnica de t-SNE para reduzir a dimensionalidade do vetor e obter a representação 2D do mesmo. Aplicando esta lógica a todas as sequências, conseguiram-se construir gráficos da representação das sequências, aplicados a uma equipa. Foram também utilizados os valores dos vetores latentes das equipas para realizar uma análise de similaridade, usando a técnica de similaridade de cosseno, obtendo assim os valores de proximidade das equipas em questão de sequências, ou, consequentemente, do estilo de jogo.

Após tudo isto e voltando à questão de investigação:

Será possível, através de técnicas de DL, a partir de dados de eventos de futebol conseguir extrair as sequências e padrões de jogo de equipas de futebol?

Os resultados apresentados no capítulo 4 permitem verificar que efetivamente é possível, a partir de dados de fluxo de eventos a construção de sequências, extração das características das mesmas e apresentar uma série de estatísticas que com a devida análise permitem potencializar os resultados desportivos da equipa que fizer proveito deles. Caso estes resultados sejam aplicados no contexto do treinador, os mesmos resultados podem ser relacionados ao treinador e tirar *insights* valiosos dos mesmos.

5.2 Objetivos Concretizados

Os objetivos deste projeto foram estabelecidos no início do projeto e mantiveram-se inalterados até à data da sua finalização.

A tabela seguinte mostra uma lista de objetivos previamente identificados e o seu estado em termos de desenvolvimento, aquando do fim do projeto.

Tabela 10: Objetivos concretizados durante o desenvolvimento do Projeto

OBJETIVO	ESTADO	OBSERVAÇÕES
Estudar e sintetizar os conhecimentos acerca de ML e DL na análise de dados, mais concretamente na área do futebol.	Feito	Completo e sintetizado no capítulo do Estado da Arte.
Construir um algoritmo que a partir de dados de eventos futebolísticos consiga retornar sequências.	Feito	Foi implementada a função e explicada no capítulo de Descrição da Implementação.
Construir um modelo que classifica as sequências, de acordo com os seus padrões.	Feito	Foi construído um autoencoder com o objetivo de extração dos padrões das sequências.
Construir um modelo de cálculo de expected goals.	Feito	
Construir um modelo de cálculo de expected threats.	Feito	
Testar e avaliar o desempenho dos modelos desenvolvidos.	Feito	Teste e avaliação sintetizado no capítulo 4.2 do presente documento.

Este projeto resultou numa solução robusta para o caso da concretização dos requisitos identificados.

5.3 Limitações e trabalho futuro

As limitações mais desafiadoras para este projeto estavam relacionadas às restrições de hardware, especialmente à limitação de memória da GPU. Isso exigiu que se fizessem algumas adaptações para lidar com essas limitações e otimizar o desempenho do sistema. Uma das

soluções adotadas foi executar o cálculo das equipas semelhantes usando apenas as sequências que terminaram em remate, reduzindo a quantidade de dados processados e economizando recursos. Essa abordagem permitiu que se focasse nas informações mais relevantes para a análise.

Além disso, outro desafio foi o tempo de treino dos modelos. Como o treino de modelos corresponde a um processo intensivo, precisou-se de encontrar maneiras de acelerar e otimizar o treino. Foram utilizadas técnicas como otimização de Hiperparâmetros para reduzir o tempo necessário para treinar os modelos, sem comprometer significativamente a sua qualidade e desempenho.

Por fim, a última limitação encontrada e já exposta foi a obtenção de serviços gratuitos que pudessem ser usados para comparação com os resultados obtidos do modelo de xT e cálculo das equipas semelhantes.

É importante ressaltar que, apesar dessas limitações, conseguiu-se superar os obstáculos e obter resultados satisfatórios. Embora se tenha enfrentado estas restrições conseguiu-se implementar uma solução eficaz e encontrar um equilíbrio entre recursos disponíveis e a complexidade do projeto. Essas experiências proporcionaram valiosas lições sobre a gestão de limitações e a procura de soluções criativas para alcançar os objetivos propostos.

Para trabalho futuro, o autor imagina um projeto que utiliza este como base e incrementa funcionalidades.

Primeiramente poderiam ser construídos novos modelos para calcular muitas outras estatísticas avançadas explicadas no Estado de Arte. Este incremento tornaria a aplicação web mais rica.

De seguida, pretende-se implementar uma página de comparação de stats, relativamente a equipas, para mais fácil comparação das mesmas. Uma página de visualização das sequências de maior xG e xT associado a uma equipa e um ranking de equipas associado a cada estatística avançada. Deste modo era possível, na primeira funcionalidade, visualizar as jogadas que mais perigo criaram e era expectável o golo e, na segunda, ordenar as equipas pelas estatísticas calculadas.

Outra alteração pretendida para o futuro seria a adaptação e visualização das métricas para o treinador da equipa. O autor acredita que o estilo de jogo está fortemente associado ao

treinador associado à equipa e deste modo, a comparação da mesma equipa em períodos de treinadores diferentes iria ser uma funcionalidade de extrema importância e utilidade e, conforme averiguado no estado de Arte, esta é uma lacuna nos serviços atualmente disponíveis no mercado.

A última implementação seria a de cache nos pedidos, de modo a tornar a resposta mais rápida, a criação de uma aplicação mobile responsiva, para ser de mais fácil acesso e o teste da aplicação web com a implementação de testes unitários e de integração.

5.4 Apreciação Final

Este projeto foi uma experiência desafiadora e gratificante. Envolveu a passagem sobre diversas áreas da inteligência artificial e o uso de diversas técnicas. Sempre tive bastante curiosidade sobre o tema, o que tornou o trabalho um pouco mais fácil. Agradeço a toda a gente que me apoio durante este projeto, em especial ao Supervisor Luís Costa, pela ajuda constante.

No entanto, nem tudo correu completamente bem. A falta de tempo e dificuldade na conciliação entre trabalho e faculdade limitou a concretização de todas as ideias que poderiam ter melhorado ainda mais a solução entregue.

Para concluir, esta foi uma experiência de aprendizagem incrível. Foi uma excelente oportunidade de pôr em prática todas as metodologias, boas práticas e conhecimentos aprendidos durante o Mestrado e que contribuíram imensamente para a concretização e sucesso do Projeto.

Referências

Akker, J. J. H. van den. (2006). *Educational design research*. 163.

Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2021). Applications of Generative Adversarial Networks (GANs): An Updated Review. *Archives of Computational Methods in Engineering*, *28*(2), 525–552. https://doi.org/10.1007/s11831-019-09388-y

Analyisport. (2021). *How Football Clubs Use Data to Sign Players - AnalyiSport*. https://analyisport.com/insights/how-football-clubs-use-data-to-sign-players/

Apache. (n.d.). *What is Airflow? — Airflow Documentation*. Retrieved June 24, 2023, from https://airflow.apache.org/docs/apache-airflow/stable/index.html

Baskerville, R. L., Kual, M., & Storey, V. C. (2015). Genres of Inquiry in Design-Science Research: Justification and Evaluation of Knowledge Production. *MIS Quarterly*, *39*(3), 541–564. https://misq.umn.edu/genres-of-inquiry-in-design-science-research-justification-and-evaluation-of-knowledge-production.html

Brechot, M., & Flepp, R. (2020). Dealing With Randomness in Match Outcomes: How to Rethink Performance Evaluation in European Club Football Using Expected Goals. *Journal of Sports Economics*, *21*(4), 335–362. https://doi.org/10.1177/1527002519897962

Brown, S. (2018). *The C4 Model for Software Architecture*. https://www.infoq.com/articles/C4-architecture-model/

CamVision. (n.d.). *Panoris*. Retrieved January 8, 2023, from https://www.panoris.com/features/

Cartas, A., Ballester, C., & Haro, G. (2022). A Graph-Based Method for Soccer Action Spotting Using Unsupervised Player Classification. *MMSports 2022 - Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*, 93–102. https://doi.org/10.1145/3552437.3555691

CAWLEY, C. (2022). *19 Awesome Uses for a Raspberry Pi*. https://www.makeuseof.com/tag/different-uses-raspberry-pi/

Chadebec, C., Vincent, L. J., & Allassonnière, S. (2022). *Pythae: Unifying Generative Autoencoders in Python -- A Benchmarking Use Case*. https://arxiv.org/abs/2206.08309v1

Champdas. (n.d.). *创冰DATA_足球数据网站| 创冰科技*. Retrieved January 8, 2023, from http://data.champdas.com/

Chandradas, A. (2021). *Getting started with Soccer Analytics with Event data | by Abhijith Chandradas | MLearning.ai | Medium*. https://medium.com/mlearning-ai/getting-started-with-soccer-analytics-with-event-data-6ecd3143e78

Chemuturi, M. (2013). Requirements Engineering and Management for Software

Development Projects. In *Requirements Engineering and Management for Software Development Projects*. Springer New York. https://doi.org/10.1007/978-1-4614-5377-2

Chen, J., & Ran, X. (2019). Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*. https://doi.org/10.1109/JPROC.2019.2921977

Cheong, C., Cheong, F., & Filippou, J. (2013). Using Design Science Research to Incorporate Gamification into Learning Activities. *PACIS 2013 Proceedings*. https://aisel.aisnet.org/pacis2013/156

Chollet, F. (2021). Deep Learning with Python, Second Edition. *Deep Learning with Python*. https://www.manning.com/books/deep-learning-with-python-second-edition

ChyronHego. (2022). *Sports Technology – ChyronHego*. https://chyronhego.com/content_tags/sports-technology/

Decroos, T., Bransen, L., Haaren, J. Van, & Davis, J. (2019). *Actions Speak Louder than Goals: Valuing Player Actions in Soccer*. *11*. https://doi.org/10.1145/3292500.3330758

Dekker, M. (2008). *File:4+1 Architectural View Model.jpg - Wikimedia Commons*. https://commons.wikimedia.org/wiki/File:4%2B1_Architectural_View_Model.jpg

Eeles, P. (2004). *Rational-Capturing Architectural Requirements Capturing Architectural Requirements*.

Everett, G., Beal, R., Matthews, T., Norman, T., & Ramchurn, G. (2022). *Contextual Expected Threat using Spatial Event Data*.

Football Analytics 101. (2019). *Football Analytics Companies & Products — Football-Analytics-101 documentation*. https://football-analytics-101.readthedocs.io/en/latest/company.html

GitHub. (n.d.). *GitHub - clementchadebec/benchmark_VAE: Unifying Variational Autoencoder (VAE) implementations in Pytorch (NeurIPS 2022)*. Retrieved June 24, 2023, from https://github.com/clementchadebec/benchmark_VAE

Goes, K. G. (2021). *Estimating the most important football player statistics using neural networks*. https://studenttheses.uu.nl/handle/20.500.12932/40763

Grenha Teixeira, J., Patrício, L., Huang, K. H., Fisk, R. P., Nóbrega, L., & Constantine, L. (2016). The MINDS Method. *Http://Dx.Doi.Org/10.1177/1094670516680033*, *20*(3), 240–258. https://doi.org/10.1177/1094670516680033

Han, J., Kamber, M., & Pei, J. (2012). Getting to Know Your Data. *Data Mining*, 39–82. https://doi.org/10.1016/B978-0-12-381479-1.00002-2

Harkins, J. (2022). *Introducing a Possessions Framework - Stats Perform*. https://www.statsperform.com/resource/introducing-a-possessions-framework/

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly: Management Information Systems*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Horváth, I. (2007). Comparison of Three Methodological Approaches of Design Research. *DS*

42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007, 361-362 (exec. Summ.), full paper no. DS42_P_341. https://www.designsociety.org/publication/25512/Comparison+of+Three+Methodological+Approaches+of+Design+Research

InStat. (2022). *Instat*. https://www.instatsport.com/en/

InStat Scout. (n.d.). *Football • Instat*. Retrieved January 8, 2023, from https://www.instatsport.com/en/football/

Kelleher, J. D. (2019). *Deep Learning*. https://doi.org/10.7551/MITPRESS/11171.001.0001

Kruchten, P. (1995). *Architectural Blueprint The "4+1" View Model of Software Architecture*.

Lapão, L. V., Da Silva, M. M., & Gregório, J. (2017). Implementing an online pharmaceutical service using design science research. *BMC Medical Informatics and Decision Making*, *17*(1). https://doi.org/10.1186/S12911-017-0428-2

Lichtenthaler, U. (2022). Mixing data analytics with intuition: Liverpool Football Club scores with integrated intelligence. *Journal of Business Strategy*, *43*(1), 10–16. https://doi.org/10.1108/JBS-06-2020-0144

Mahesh, B. (2018). Machine Learning Algorithms-A Review. *International Journal of Science and Research*. https://doi.org/10.21275/ART20203995

Majumdar, A., Bakirov, R., Hodges, D., Scott, S., & Rees, T. (2022). Machine Learning for Understanding and Predicting Injuries in Football. *Sports Medicine - Open*, *8*(1), 1–10. https://doi.org/10.1186/S40798-022-00465-4/TABLES/3

McKenney, S. E., & Reeves, T. C. (Thomas C. (2012). *Conducting educational design research*. https://www.routledge.com/Conducting-Educational-Design-Research/McKenney-Reeves/p/book/9781138095564

Merhej, C., Beal, R., Ramchurn, S., & Matthews, T. (2021). What Happened Next? Using Deep Learning to Value Defensive Actions in Football Event-Data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *21*, 3394–3403. https://doi.org/10.1145/3447548.3467090

Molina, M., Castro, E., & Castro, E. (2007). Teaching experiments within design research. *International Journal of Interdisciplinary Social Sciences*, *2*(4), 435–440. https://doi.org/10.18848/1833-1882/CGP/V02I04/52362

Nielsen, J. (1993). *Usability Engineering*. California: SunSoft.

OPTA Documentation. (n.d.). *Opta Playground- F24 documentation*. Retrieved June 24, 2023, from https://studylib.net/doc/25339973/opta-playground--f24-documentation

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, *372*. https://doi.org/10.1136/bmj.n71

Pantzalis, V. C., & Tjortjis, C. (2020). Sports Analytics for Football League Table and Player Performance Prediction. *11th International Conference on Information, Intelligence, Systems and Applications, IISA 2020*. https://doi.org/10.1109/IISA50023.2020.9284352

Paraeles, E. (2019). *A evolução das transferências no mercado do futebol*. https://www.paraeles.pt/desporto/evolucao-transferencias-futebol/

ParseHub. (2023). *What is Web Scraping and What is it Used For? | ParseHub*. https://www.parsehub.com/blog/what-is-web-scraping/

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pimentel, M., & Filippo, D. (2020). *RE@D-Revista de Educação a Distância e Elearning Design Science Research: pesquisa científica atrelada ao design de artefatos*.

PIXINVENT. (n.d.). *Vuexy - Vuejs, React - Next.js, HTML, Laravel & Asp.Net Admin Dashboard Template by PIXINVENT*. Retrieved June 27, 2023, from https://themeforest.net/item/vuexy-vuejs-html-laravel-admin-dashboard-template/23328599?gclid=CjwKCAjwkeqkBhAnEiwA5U-uMwFnLZoNcP2Gpm7hD0lN0Ot0ewlmJD2VFgDHA7hcG7nIdFI0d3KupxoCuzsQAvD_BwE

Pratik, S., & Iriondo, R. (2022). *Main Types of Neural Networks and its Applications — Tutorial – Towards AI*. https://towardsai.net/p/machine-learning/main-types-of-neural-networks-and-its-applications-tutorial-734480d7ec8e

Pulis, M., & Bajada, J. (2022). *Reinforcement Learning for Football Player Decision Making Analysis*.

Ramos, C. (2020). *6 – Introdução à Aprendizagem Profunda (Deep Learning) Onde se situa a Aprendizagem Profunda ? 2*.

Rethlefsen, M. L., Kirtley, S., Waffenschmidt, S., Ayala, A. P., Moher, D., Page, M. J., Koffel, J. B., Blunt, H., Brigham, T., Chang, S., Clark, J., Conway, A., Couban, R., de Kock, S., Farrah, K., Fehrmann, P., Foster, M., Fowler, S. A., Glanville, J., … Group, P.-S. (2021). PRISMA-S: an extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews. *Systematic Reviews*, *10*(1), 39. https://doi.org/10.1186/s13643-020-01542-z

Roy, A. (2020). *Introduction To Autoencoders. A Brief Overview | by Abhijit Roy | Towards Data Science*. https://towardsdatascience.com/introduction-to-autoencoders-7a47cf4ef14b

Roy, M. Van, Robberechts, P., Decroos, T., & Davis, J. (2020). *Valuing On-the-Ball Actions in Soccer: A Critical Comparison of xT and VAEP*. www.aaai.org

SAP Sports One. (n.d.). *Sports Team Management Software | SAP Sports One*. Retrieved January 8, 2023, from https://www.sap.com/products/technology-platform/sports-one.html

SBE, N. (2021). *Ciência de dados no futebol permite aos clubes ganhar «escala e eficiência»*. https://www.novasbe.unl.pt/en/faculty-research/knowledge-centers/data-science/news/news-detail-data-science/id/576/ciencia-de-dados-no-futebol-permite-

aos-clubes-ganhar-escala-e-eficiencia

SciSports. (2021). *State of the football analytics industry in 2021 - SciSports*. https://www.scisports.com/state-of-the-football-analytics-industry-in-2021/

Selenium. (n.d.). *Selenium*. Retrieved June 24, 2023, from https://www.selenium.dev/

Singh, K. (2019). *Introducing Expected Threat (xT)*. https://karun.in/blog/expected-threat.html

SkillCorner. (n.d.). *A New Dimension In Football Analytics | SkillCorner*. Retrieved January 8, 2023, from https://www.skillcorner.com/#home-page

SKLearn Metrics. (n.d.). *sklearn.metrics.brier_score_loss — scikit-learn 1.2.2 documentation*. Retrieved June 24, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.brier_score_loss.html

Smith, C. U., & Williams, L. G. (2003). *Best Practices for Software Performance Engineering*. http://www.perfeng.com/

Soccerment. (2022a). *Soccerment's Advanced Metrics | Soccerment Research*. https://soccerment.com/soccerments-advanced-metrics/

Soccerment. (2022b). *The Very Exclusive xOVA Club | Soccerment Research*. https://soccerment.com/the-very-exclusive-xova-club/

Sportinforma. (2019). *A evolução da "loucura" no mercado de transferências: de Maradona a Figo, passando por Ronaldo, Félix e Neymar - Futebol - SAPO Desporto*. https://desporto.sapo.pt/futebol/artigos/a-evolucao-da-loucura-no-mercado-de-transferencias-de-maradona-a-figo-passando-ronaldo-felix-e-neymar

Stats LLC. (2022). *Stats Perform Playing Styles – An Introduction - Stats Perform*. https://www.statsperform.com/resource/stats-playing-styles-introduction/

Stats Perform. (2022a). *Artificial Intelligence - Stats Perform*. https://www.statsperform.com/artificial-intelligence/

Stats Perform. (2022b). *Opta data from Stats Perform*. https://www.statsperform.com/opta/

Stats Perform. (2022c). *Opta Predictions - Stats Perform*. https://www.statsperform.com/opta-predictions/

StatsBomb. (n.d.-a). *IQ Soccer | Soccer Data Analytics Platform | StatsBomb*. Retrieved January 8, 2023, from https://statsbomb.com/what-we-do/iq-soccer/

StatsBomb. (n.d.-b). *Who We Are | About Us | StatsBomb*. Retrieved January 8, 2023, from https://statsbomb.com/who-we-are/

StatsBomb. (2022a). *StatsBomb | Data Champions*. https://statsbomb.com/

StatsBomb. (2022b). *What is xG? How is it calculated? | StatsBomb | Data Champions*. https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/

The Punters Page. (2023). ► *Expected Goals (xG) Explained • Easy Guide To xG Football Stats!* •. https://www.thepunterspage.com/expected-goals-explained/

The xG Philosophy. (2020). *The xG Philosophy no Twitter: "How does the Expected Goals method deal with two consecutive high quality shots in quick succession? THREAD:- (1/6) https://t.co/sgqnFme8ly" / Twitter*. https://twitter.com/xGPhilosophy/status/1285638896089522177

Todinov, M. (2016). *Reliability and Risk Models: Setting Reliability Requirements* (Second ed.). John Wiley & Sons, Ltd.

Tongdaoweiye. (n.d.). 同道伟业. Retrieved January 8, 2023, from http://www.cfadata.cn/#/

Track160. (2021). *Optical tracking: The holy grail of tracking data In football*. https://www.track160.com/post/optical-tracking-the-holy-grail-of-tracking-data-in-football

Tripathy, S. D. (2022). *xOVA Explained: The Metric That Changed Football Data Analytics*. https://www.foottheball.com/explainer/explained-what-is-xova-ova-stats-numbers-data-analytics-football-soccerment/

União Europeia. (2016). *Regulamento Geral sobre a Proteção de Dados*.

Vidal-Codina, F., Evans, N., El Fakir, B., & Billingham, J. (2022). Automatic event detection in football using tracking data. *Sports Engineering*, *25*(1), 1–15. https://doi.org/10.1007/S12283-022-00381-6/FIGURES/10

Violante, A. (2018). *An Introduction to t-SNE with Python Example | by Andre Violante | Towards Data Science*. https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1

Whitmore, J. (2021a). *Sequences and Possessions in Football | The Analyst*. https://theanalyst.com/eu/2021/03/possessions-and-sequences-in-football/

Whitmore, J. (2021b). *What Are Expected Assists (xA)? | The Analyst*. https://theanalyst.com/eu/2021/03/what-are-expected-assists-xa/

Whitmore, J. (2021c). *What Are Expected Goals (xG)? | The Analyst*. https://theanalyst.com/eu/2021/07/what-are-expected-goals-xg/

Whitmore, J. (2022). *Evolving Expected Goals (xG) | The Analyst*. https://theanalyst.com/eu/2022/04/evolving-expected-goals-xg/

Wijngaard, G. W. A. (2020). *Clustering soccer players: investigating unsupervised learning on player positions*. https://studenttheses.uu.nl/handle/20.500.12932/35795

Worville, T. (2022). *Expected assists in context - Stats Perform*. https://www.statsperform.com/resource/expected-assists-in-context-2/

Wyscout. (2022). *Professional Football Platform for Football Analysis - Wyscout*. https://wyscout.com/

Zhou, Z.-H. (2021). *Machine Learning*. https://doi.org/10.1007/978-981-15-1967-3

Anexo

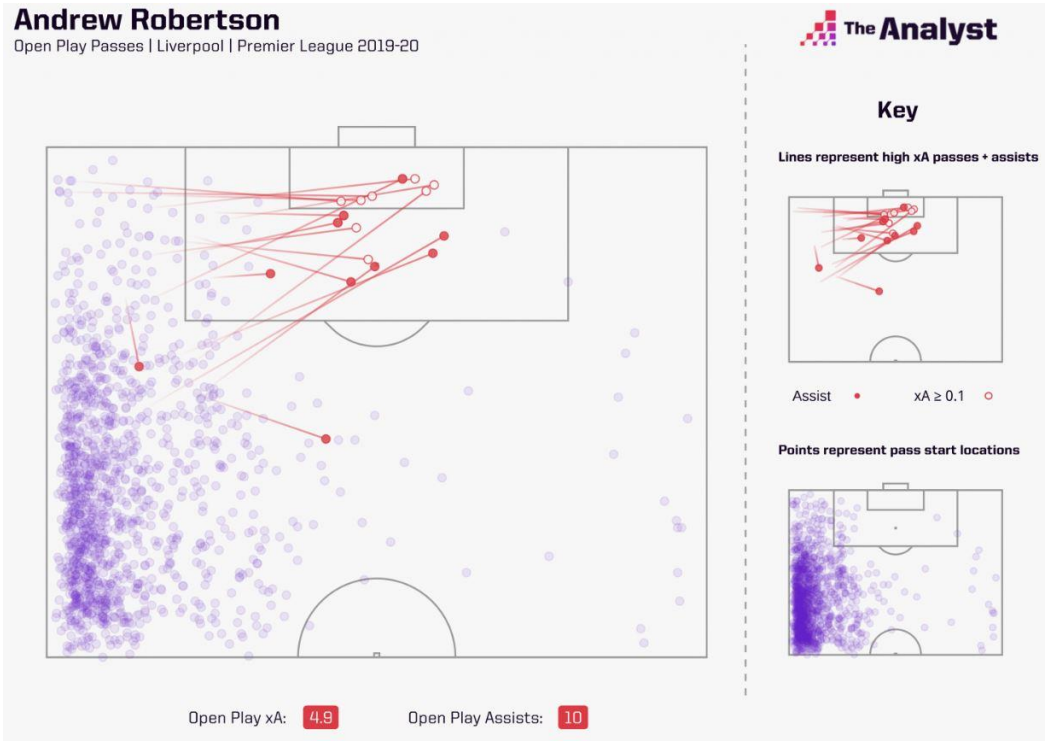


Figura 70: Mapa de xA de Andrew Robertson para a época 2019-20 (Whitmore, 2021b)

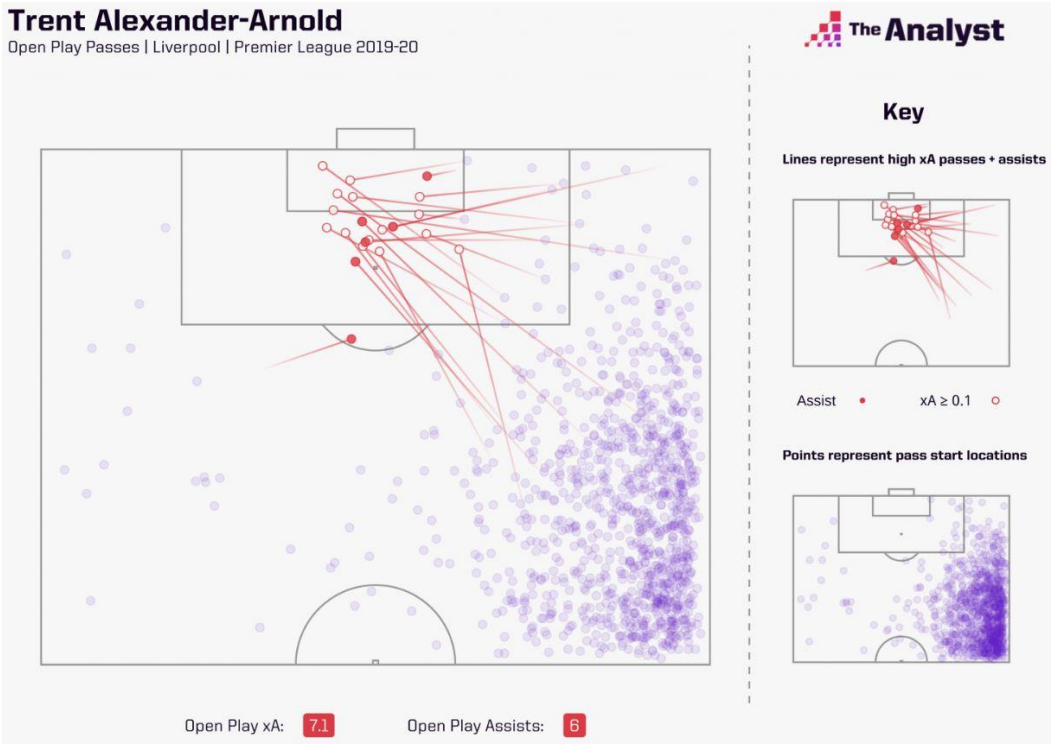


Figura 71: Mapa de xA de Alexander Arnold para a época 2019-20 (Whitmore, 2021b)

```
1.
2.  for seq in seqs:
3.          seq_df = sequences_match.loc[sequences_match['sequence_id'] == seq]
4.
5.          seq_df.head(1)['ball_receiving_x'] = np.nan
6.          seq_df.head(1)['ball_receiving_y'] = np.nan
7.          seq_df = seq_df.loc[seq_df['result_name'] == 'success']
8.
9.          if seq_df.shape[0] < 2:
10.             continue
11.
12.         image_df = seq_df.copy()
13.
14.         image_df.loc[image_df['type_name'] == 'throw_in', 'ball_receiving_x'] =
    image_df.loc[image_df['type_name'] == 'throw_in', 'start_x']
15.         image_df.loc[image_df['type_name'] == 'throw_in', 'ball_receiving_y'] =
    image_df.loc[image_df['type_name'] == 'throw_in', 'start_y']
16.
17.         image_df['start_x'] = [desired_x_bounds[0] + (x - actual_x_bounds[0]) *
    (desired_x_bounds[1] - desired_x_bounds[0]) / (actual_x_bounds[1] -
    actual_x_bounds[0]) for x in image_df.start_x.values]
18.         image_df['end_x'] = [desired_x_bounds[0] + (x - actual_x_bounds[0]) *
    (desired_x_bounds[1] - desired_x_bounds[0]) / (actual_x_bounds[1] -
    actual_x_bounds[0]) for x in image_df.end_x.values]
19.
20.         image_df['start_y'] = [desired_y_bounds[0] + (x - actual_y_bounds[0]) *
    (desired_y_bounds[1] - desired_y_bounds[0]) / (actual_y_bounds[1] -
    actual_y_bounds[0]) for x in image_df.start_y.values]
21.         image_df['end_y'] = [desired_y_bounds[0] + (x - actual_y_bounds[0]) *
    (desired_y_bounds[1] - desired_y_bounds[0]) / (actual_y_bounds[1] -
    actual_y_bounds[0]) for x in image_df.end_y.values]
22.
23.         image_df['ball_receiving_y'] = [desired_y_bounds[0] + (x -
    actual_y_bounds[0]) * (desired_y_bounds[1] - desired_y_bounds[0]) /
    (actual_y_bounds[1] - actual_y_bounds[0]) for x in
    image_df.ball_receiving_y.values]
24.         image_df['ball_receiving_x'] = [desired_x_bounds[0] + (x -
    actual_x_bounds[0]) * (desired_x_bounds[1] - desired_x_bounds[0]) /
    (actual_x_bounds[1] - actual_x_bounds[0]) for x in
    image_df.ball_receiving_x.values]
25.
26.         final_image_df = pd.DataFrame(columns=['start_x', 'end_x', 'start_y',
    'end_y'])
27.         for i, row in image_df.iterrows():
28.             ball_receiving_y = row['ball_receiving_y']
29.             ball_receiving_x = row['ball_receiving_x']
30.             start_x = row['start_x']
31.             start_y = row['start_y']
32.             end_x = row['end_x']
33.             end_y = row['end_y']
34.
35.             row = [ball_receiving_x, start_x, ball_receiving_y, start_y]
36.             final_image_df.loc[-1] = row
37.             final_image_df.index = final_image_df.index + 1
38.             final_image_df.sort_index()
39.
40.             row = [start_x, end_x, start_y, end_y]
41.
42.             final_image_df.loc[-1] = row
43.             final_image_df.index = final_image_df.index + 1
44.             final_image_df.sort_index()
45.
46.
47.         final_image_df =
    final_image_df.dropna().sort_index(ascending=False).reset_index(drop=True)
48.         # Create a black image
49.         img = np.zeros((IMAGE_SIZE_X,IMAGE_SIZE_Y,1), np.uint8)
```

```
        step = 5
        for i,row in final_image_df.iterrows():

    cv.line(img,(int(row['start_y']),int(row['start_x'])),(int(row['end_y']),int(row['
    end_x'])),(255-(i*step),255-(i*step),255-(i*step)),2)

        img = cv.flip(img, 0)
        img = cv.flip(img, 1)

        path = f"dataset\\{IMAGE_SIZE_X}x{IMAGE_SIZE_Y}"
        if os.path.exists(path) == False:
            os.makedirs(path)

        cv.imwrite(f"{path}\\{match}_{seq_df.iloc[0].team_id}_{seq}.png", img)
```

Listagem 23: Construção das Imagens

```

class Encoder(nn.Module):
    def __init__(self, latent_dim=16):
        super(Encoder, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 3, stride=2, padding=1)
        self.conv2 = nn.Conv2d(32, 64, 3, stride=2, padding=1)
        self.conv3 = nn.Conv2d(64, 128, 3, stride=2, padding=1)
        self.conv4 = nn.Conv2d(128, 256, 3, stride=2, padding=1)
        self.fc_mu = nn.Linear(256*16*16, latent_dim)
        self.fc_logvar = nn.Linear(256*16*16, latent_dim)

    def forward(self, x):
        x = nn.functional.relu(self.conv1(x))
        x = nn.functional.relu(self.conv2(x))
        x = nn.functional.relu(self.conv3(x))
        x = nn.functional.relu(self.conv4(x))
        x = x.view(x.size(0), -1)
        mu = self.fc_mu(x)
        logvar = self.fc_logvar(x)
        return mu, logvar


class Decoder(nn.Module):
    def __init__(self, latent_dim=16):
        super(Decoder, self).__init__()
        self.fc = nn.Linear(latent_dim, 256*16*16)
        self.conv1 = nn.ConvTranspose2d(256, 128, 4, stride=2, padding=1)
        self.conv2 = nn.ConvTranspose2d(128, 64, 4, stride=2, padding=1)
        self.conv3 = nn.ConvTranspose2d(64, 32, 4, stride=2, padding=1)
        self.conv4 = nn.ConvTranspose2d(32, 1, 4, stride=2, padding=1)

    def forward(self, z):
        x = self.fc(z)
        x = x.view(x.size(0), 256, 16, 16)
        x = nn.functional.relu(self.conv1(x))
        x = nn.functional.relu(self.conv2(x))
        x = nn.functional.relu(self.conv3(x))
        x = torch.sigmoid(self.conv4(x))
        return x


class VAE(nn.Module):
    def __init__(self, latent_dim=16):
        super(VAE, self).__init__()
        self.latent_dim = latent_dim
        self.encoder = Encoder(latent_dim)
```

```
        self.decoder = Decoder(latent_dim)

    def reparameterize(self, mu, logvar):
        std = torch.exp(0.5 * logvar)
        eps = torch.randn_like(std)
        return eps * std + mu

    def forward(self, x):
        mu, logvar = self.encoder(x)
        z = self.reparameterize(mu, logvar)
        x_hat = self.decoder(z)
        return x_hat, mu, logvar
```

Listagem 24: Versão do VAE construído para imagens de 256x256

```

def train_epoch(model, device, dataloader, loss_fn, optimizer):
    # Set train mode for both the encoder and the decoder
    model.train()
    train_loss = []
    # Iterate the dataloader (we do not need the label values, this is unsupervised learning)
    for image_batch, _ in dataloader: # with "_" we just ignore the labels (the second element of the dataloader tuple)
        # Move tensor to the proper device
        image_batch = image_batch.to(device)
        optimizer.zero_grad()
        x_hat, mu, logvar = model(image_batch)
        reproduction_loss,eval_loss  = loss_fn(image_batch, x_hat, mu, logvar)
        # Backward pass
        reproduction_loss.backward()
        optimizer.step()
        # Print batch loss
        print('\t partial train loss (single batch): %f' % (eval_loss.data))
        train_loss.append(eval_loss.detach().cpu().numpy())
    return np.mean(train_loss)

```

Listagem 25: Função de Treino de uma Época

```

def test_epoch(model, device, dataloader, loss_fn):
    # Set evaluation mode for encoder and decoder
    model.eval()
    with torch.no_grad(): # No need to track the gradients
        # Define the lists to store the outputs for each batch
        conc_out = []
        conc_label = []
        conc_mu = []
        conc_logvar = []
        for image_batch, _ in dataloader:
            # Move tensor to the proper device
            image_batch = image_batch.to(device)
            decoded_data, mu, logvar = model(image_batch)
            # Append the network output and the original image to the lists
            conc_out.append(decoded_data.cpu())
            conc_label.append(image_batch.cpu())
            conc_mu.append(mu)
            conc_logvar.append(logvar)
        # Create a single tensor with all the values in the lists
        conc_out = torch.cat(conc_out)
        conc_label = torch.cat(conc_label)
        conc_mu = torch.cat(conc_mu)
```

```
        conc_logvar = torch.cat(conc_logvar)
        # Evaluate global loss
        repro_loss, eval_loss = loss_fn(conc_out, conc_label,conc_mu,conc_logvar )
    return eval_loss.data
```

Listagem 26: Função de Teste de uma Época

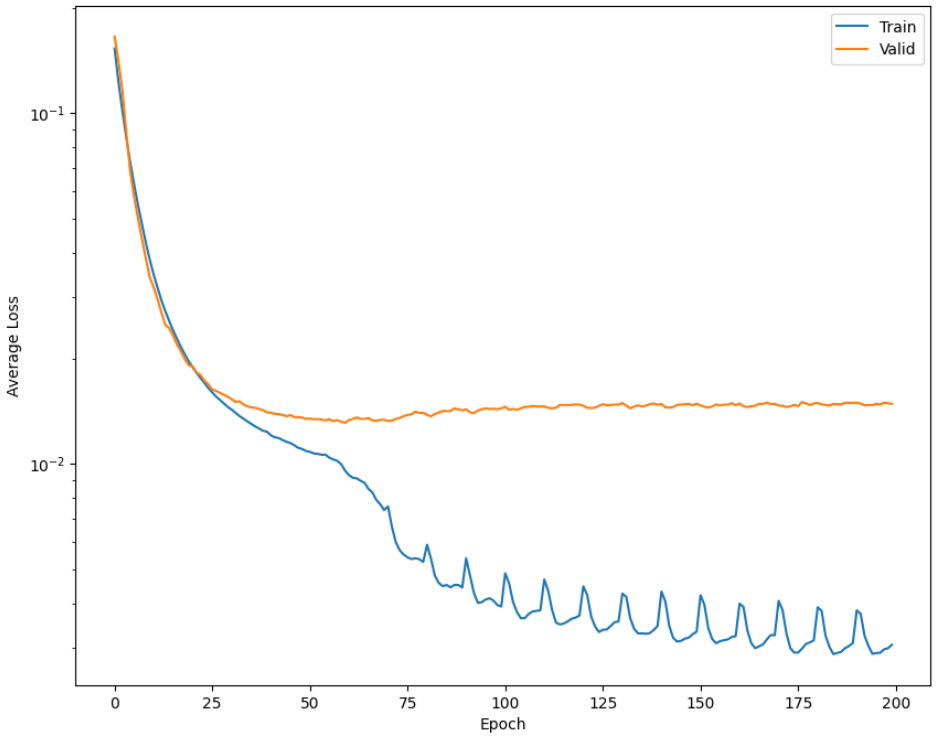


Figura 72: Gráficos do Treino do Modelo MSE em função do tempo para Autoencoder Base

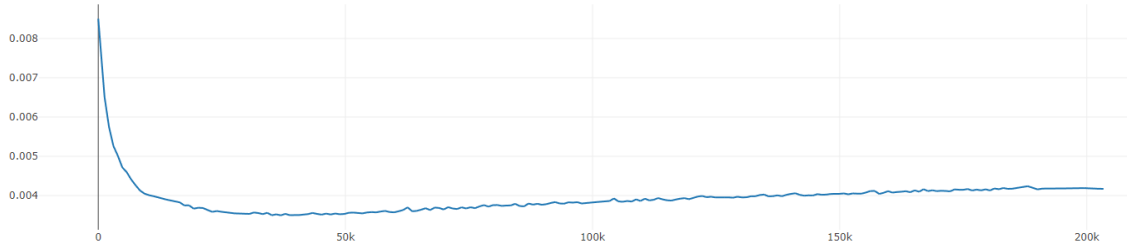


Figura 73: Gráficos do Treino do Modelo MSE em função do tempo para RAE-L2

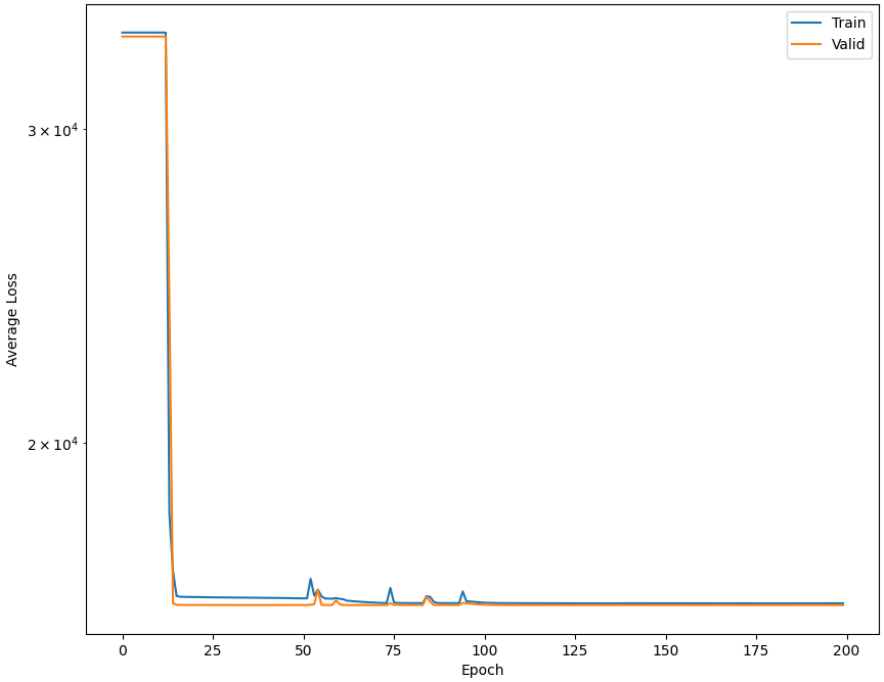


Figura 74: Gráficos do Treino do Modelo para VAEGAN

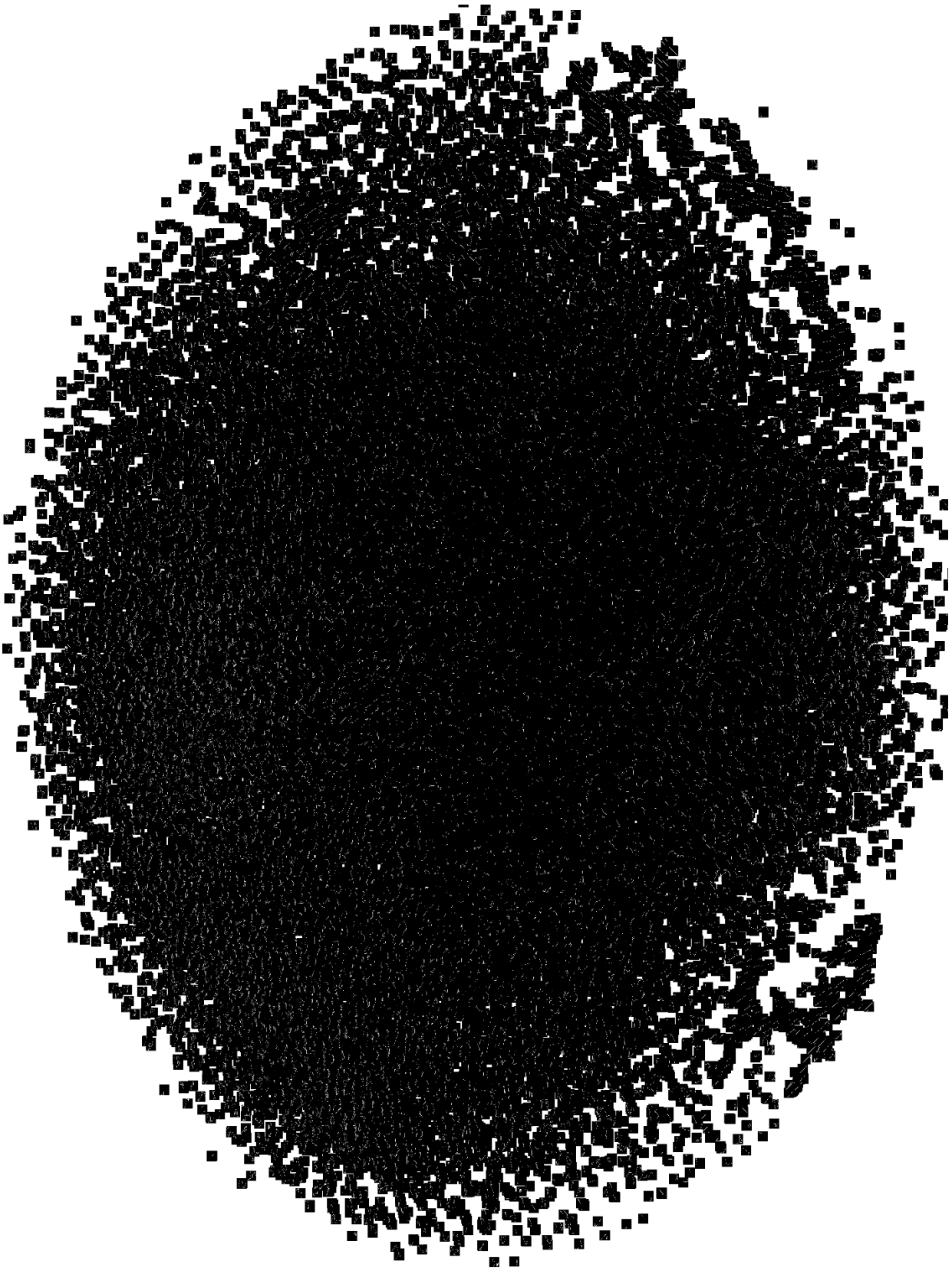


Figura 75: Representação do t-SNE em todas as sequências

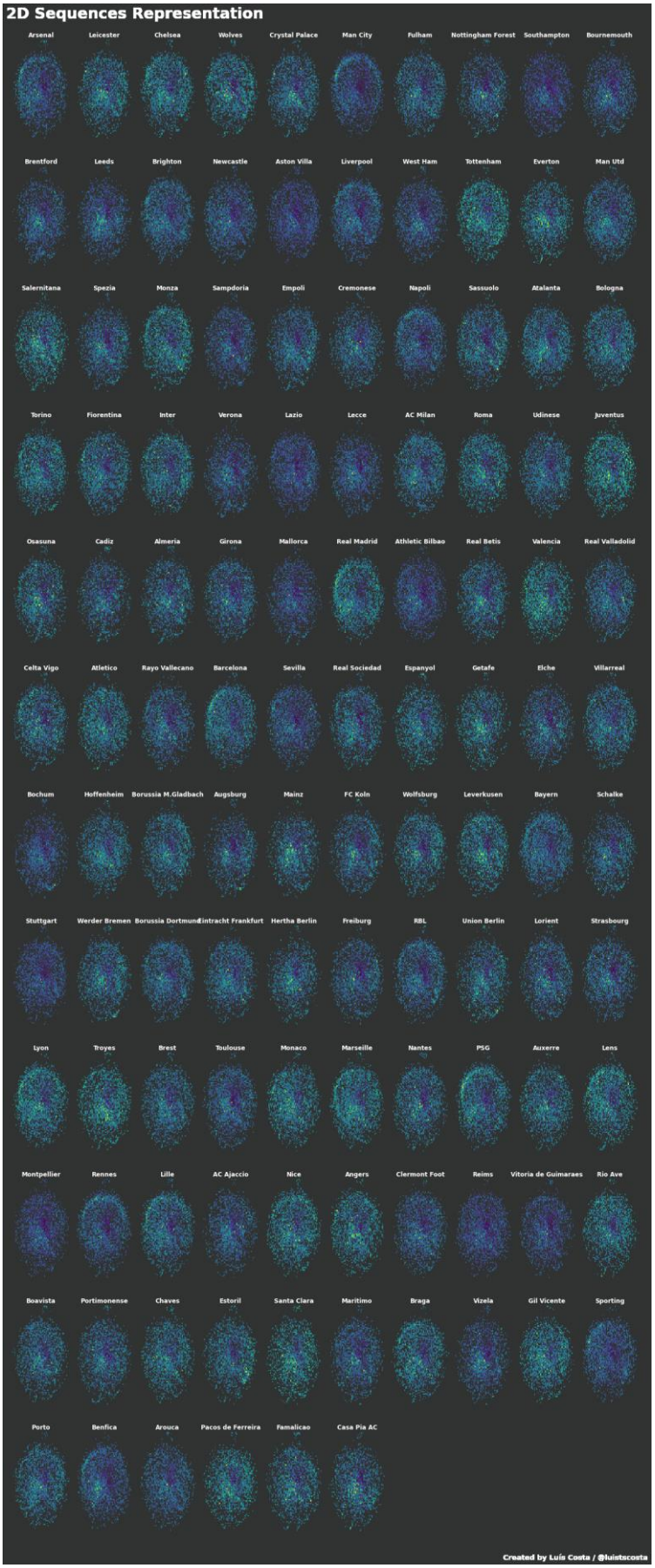

2D Sequences Representation
Arsenal
Leicester
Chelsea
Wolves
Crystal Palace
Man City
Fulham
Nottingham Forest
Southampton
Bournemouth
Brentford
Leeds
Brighton
Newcastle
Aston Villa
Liverpool
West Ham
Tottenham
Everton
Man Utd
Salernitana
Spezia
Monza
Sampdoria
Empoli
Cremonese
Napoli
Sassuolo
Atalanta
Bologna
Torino
Fiorentina
Inter
Verona
Lazio
Lecce
AC Milan
Roma
Udinese
Juventus
Osasuna
Cadiz
Almeria
Girona
Mallorca
Real Madrid
Athletic Bilbao
Real Betis
Valencia
Real Valladolid
Celta Vigo
Atletico
Rayo Vallecano
Barcelona
Sevilla
Real Sociedad
Espanyol
Getafe
Elche
Villarreal
Bochum
Hoffenheim
Borussia M.Gladbach
Augsburg
Mainz
FC Koln
Wolfsburg
Leverkusen
Bayern
Schalke
Stuttgart
Werder Bremen
Borussia Dortmund
Eintracht Frankfurt
Hertha Berlin
Freiburg
RBL
Union Berlin
Lorient
Strasbourg
Lyon
Troyes
Brest
Toulouse
Monaco
Marseille
Nantes
PSG
Auxerre
Lens
Montpellier
Rennes
Lille
AC Ajaccio
Nice
Angers
Clermont Foot
Reims
Vitoria de Guimaraes
Rio Ave
Boavista
Portimonense
Chaves
Estoril
Santa Clara
Maritimo
Braga
Vizela
Gil Vicente
Sporting
Porto
Benfica
Arouca
Pacos de Ferreira
Famalicao
Casa Pia AC
Created by Luís Costa / @luistscosta

Figura 76: Representação do t-SNE para todas as equipas sobre a média da Liga