

Desenvolvimento de Interface para Análise de Performance Individual e em Equipe de Jogadores de Basquete

Breno Varela de Oliveira

PROJETO FINAL DE GRADUAÇÃO

**CENTRO TÉCNICO CIENTÍFICO - CTC
DEPARTAMENTO DE INFORMÁTICA
Curso de Graduação em Engenharia da
Computação**

Rio de Janeiro
Novembro de 2024



Breno Varela de Oliveira

**Desenvolvimento de Interface para Análise de
Performance Individual e em Equipe de
Jogadores de Basquete**

Relatório de Projeto Final, apresentado ao Programa de Engenharia da Computação, do Departamento de Informática da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Prof. Waldemar Celes Filho

Rio de Janeiro
Novembro de 2024

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

Breno Varela de Oliveira

Graduando em Engenharia da Computação na PUC-Rio.

Ficha Catalográfica

Varela de Oliveira, Breno

Desenvolvimento de Interface para Análise de Performance Individual e em Equipe de Jogadores de Basquete / Breno Varela de Oliveira; orientador: Waldemar Celes Filho. – 2024.

42 f: il. color. ; 30 cm

Projeto Final - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2024.

Inclui bibliografia

1. Informática – Trabalho de Conclusão de Curso. 2. Visualização de Dados. 3. Análise de Dados. 4. Basquete. 5. Sports Analytics. I. Celes Filho, Waldemar. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para minha família e amigos, que tanto me apoiaram durante essa trajetória.

Agradecimentos

Para todos os professores da PUC-Rio que marcaram minha formação acadêmica, em especial ao professor Waldemar Celes, por me orientar neste trabalho.

Para minha família, que me deu todo o suporte financeiro e emocional para alcançar essa conquista.

Para todos os amigos que fiz durante 5 anos de faculdade, por todos bons e maus momentos vividos.

Resumo

Varela de Oliveira, Breno; Celes Filho, Waldemar. **Desenvolvimento de Interface para Análise de Performance Individual e em Equipe de Jogadores de Basquete**. Rio de Janeiro, 2024. 42p. Projeto Final – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A análise de dados desempenha um papel crucial nos esportes modernos, fornecendo insights valiosos que podem influenciar estratégias e decisões. No basquete, o uso de estatísticas avançadas permite uma compreensão profunda do desempenho da equipe e também de cada jogador, ajudando a identificar pontos fortes e áreas de melhoria. Neste trabalho, será desenvolvida uma interface para o software BIRD, focada na análise de performance individual de jogadores de basquete. Com essa interface, técnicos e comissões técnicas podem avaliar o impacto de cada atleta no time, otimizando treinamentos e táticas de jogo. A ferramenta oferece uma visualização detalhada de métricas essenciais, facilitando a tomada de decisões informadas e potencializando o rendimento geral da equipe.

Palavras-chave

Visualização de Dados; Análise de Dados; Basquete; Sports Analytics.

Abstract

Varela de Oliveira, Breno; Celes Filho, Waldemar (Advisor). **Development of an Interface for Basketball Player's Performance Analysis in the BIRD Statistics Software**. Rio de Janeiro, 2024. 42p. Projeto Final – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Data analysis plays a crucial role in modern sports, providing valuable insights that can influence strategies and decisions. In basketball, the use of advanced sports analytics allows for a deep understanding of the team and also each player's performance, helping to identify strengths and areas for improvement. In this work, an interface will be developed for the BIRD software, focused on individual player performance analysis in basketball. With this interface, coaches and technical staff can evaluate the impact of each athlete on the team, optimizing training and game tactics. The tool offers a detailed visualization of essential metrics, facilitating informed decision-making and enhancing the overall performance of the team.

Keywords

Data Viz; Data Analysis; Basketball; Sports Analytics.

Sumário

1	Introdução	13
2	Situação Atual	15
3	Objetivos	17
4	Atividades Realizadas	18
4.1	Estudo sobre os Four Factors de Dean Oliver	18
4.2	Estudo sobre a API BIRD	21
4.3	Protótipos de Gráficos em Python	24
5	Projeto e Especificação do Sistema	26
5.1	Visão Geral do Sistema	26
5.2	Arquitetura do Sistema	26
5.3	Processamento de Dados	30
6	Implementação e Avaliação	33
6.1	Validação das Funcionalidades da Interface	33
6.2	Validação das Fórmulas	36
6.3	Testes em Tempo Real	37
7	Consideração Finais	39
7.1	Contribuições do Trabalho	39
7.2	Aprendizado com o Trabalho	39
7.3	Limitações e Oportunidades de Melhoria	40
7.4	Recomendações para Trabalhos Futuros	40
8	Referências bibliográficas	42

Lista de figuras

Figura 4.1	Linha dos 3 pontos	19
Figura 4.2	Rotas de Ligas	22
Figura 4.3	Rotas de Times	22
Figura 4.4	Rotas de Jogadores	23
Figura 4.5	Rotas de Partidas	23
Figura 4.6	Rotas de Eventos	24
Figura 4.7	Pontuação de um jogador durante a partida	25
Figura 5.1	Interface desenvolvida para o BIRD	27
Figura 5.2	Interface BAICA	27
Figura 5.3	Select Estilizado com Bootstrap	29
Figura 5.4	Selecionando Um Intervalo	30
Figura 5.5	Aplicação Sincronizada do Intervalo	30
Figura 6.1	Modal de Legenda	34
Figura 6.2	Desalinhamento entre os gráficos devido aos rótulos do eixo y	35
Figura 6.3	Barra de referência de tempo entre os gráficos	35
Figura 6.4	Transmissão da partida entre Brasília e Unifacisa ao Vivo	37
Figura 6.5	Interface durante partida entre Brasília e Unifacisa ao Vivo	38

Lista de algoritmos

Algoritmo 1 Processamento de dados e cálculo dos Four Factors 32

Lista de Códigos

Código 1 Interfaces Criadas

28

Lista de Abreviaturas

NBA – *National Basketball Association*

NBB – *Novo Basquete Brasil*

BAICA – *Basketball Artificial Intelligence Coach Assistant*

BIRD – *Basketball Intelligence Research Dashboard*

API – *Application Programming Interface*

CRUD – *Creation, Reading, Updating, and Deleting of data*

eFG % – *Effective Field Goal Percentage*

TOV % – *Turnover Percentage*

DRB % – *Defensive Rebound Percentage*

ORB % – *Offensive Rebound Percentage*

FTRq – *Free Throw Percentage*

O acaso favorece a mente preparada.

Louis Pasteur, *Discours de réception de Louis Pasteur à l'Académie Française (1882)*.

1

Introdução

O basquete tem se tornado um esporte cada vez mais relevante no Brasil. De acordo com matéria publicada online pela revista *Veja*, o público brasileiro que assiste à NBA, a liga de basquete americana que concentra a elite do basquete mundial, triplicou no período entre 2013 a 2023, passando de 20 para 60 milhões de espectadores (VEJA, 2023). Embora o basquete nacional não tenha tido a mesma repercussão da liga americana, o simples crescimento da base de fãs já é um fenômeno importante para o desenvolvimento do esporte e profissionalização no país.

Quando falamos de profissionalização no esporte, um dos principais temas é a utilização de análise de dados e estatísticas. Com o surgimento de novas tecnologias e avanços significativos no poder computacional, foi possível capturar, cada vez mais, dados e métricas para serem utilizados em análises estatísticas de equipes e atletas profissionais. Essa tendência de tornar o esporte profissional cada vez mais orientado a dados, começou a se destacar a partir de 2002, quando Billy Beane, na época general manager da equipe de baseball Oakland Athletics, utilizou essa técnica para formar um time de baseball competitivo com recursos financeiros extremamente limitados. Essa história foi retratada no livro *Moneyball: The Art of Winning an Unfair Game* em 2003 e em 2011 foi levada aos cinemas no filme *Moneyball* com Brad Pitt interpretando Billy, deixando o assunto ainda mais em destaque. No basquete não foi muito diferente.

Em 2004, Dean Oliver publicou o livro *Basketball on Paper* que para muitos é tido como o primeiro livro de análise de dados no basquete. Neste livro ele identifica o que ele chama de "Four Factors for Basketball Success": um conjunto de quatro métricas que, segundo ele, tem um potencial determinante no vencedor da partida. Desde então, a coleta de dados voltados para análises estatísticas teve uma evolução exponencial e, atualmente, praticamente cada passo dado em quadra durante um jogo da NBA é registrado e analisado, a fim de obter uma avaliação mais clara e precisa da performance da equipe e dos seus jogadores.

No Brasil, o basquete também tem evoluído bastante nesse aspecto, fortemente impulsionado por essa nova onda de fãs do esporte, que reflete em maior visibilidade e, conseqüentemente, maiores investimentos. Tendo em vista esse movimento, em 2020 o Clube de Regatas Flamengo e a PUC-Rio assinam um Programa de Cooperação Técnica para desenvolver seus esportes

olímpicos e o primeiro projeto conjunto foi o BAICA (*Basketball Artificial Intelligence Coach Assistant*) voltado para análise de dados em tempo real durante partidas de basquete. Esse projeto deu origem ao BIRD (*Basketball Intelligence Research Dashboard*), uma plataforma desenvolvida pelo TecGraf em parceria com a Sports Power AI, para pesquisas voltadas ao basquete e que atendem às demandas do programa BAICA.

Até o momento, a plataforma abrange os principais aspectos e estatísticas para avaliar a performance do elenco como um todo e também de cada quinteto dentro de quadra, porém carece de informações que permitam avaliar mais profundamente o desempenho individual de cada jogador. Esse tipo de análise é fundamental para a elaboração de um planejamento estratégico e individualizado de treinamentos e preparação física, alinhado às especificidades e à performance de cada atleta. Diante disso, este trabalho de conclusão de curso pretende apresentar as principais métricas de avaliação de jogadores e elaborar um dashboard para auxiliar nas tomadas de decisão do técnico e sua comissão.

2

Situação Atual

A análise de dados de basquete parte inicialmente da coleta e estruturação de dados referentes a cada evento que ocorre dentro de quadra. Esse processo é realizado pela NBB e disponibilizado para cada equipe, de modo que possam avaliar as estatísticas de seu próprio time e de seus adversários. Esses dados brutos são armazenados pela equipe do Flamengo e através deles, foi desenvolvida pelo TecGraf uma API para consulta de dados e construção das análises que são implementadas no BIRD. Uma API consiste em uma série de regras e protocolos que definem a comunicação entre o software e os dados. Essas operações são conhecidas como CRUD (*Creation, Reading, Updating, and Deleting of data*) e estabelecem uma camada entre os desenvolvedores e os dados brutos definindo permissões e o modo de interação com os dados (ORACLE, 2024).

No basquete as principais métricas analisadas são número de pontos, rebotes (quando um jogador recupera a bola após um arremesso que bateu no aro ou na tabela), assistências, arremessos, bolas recuperadas (quando a defesa intercepta um passe ou rouba a bola de um jogador, resultando em uma inversão de posse), tocos (quando um jogador bloqueia um arremesso) e erros. Essas estatísticas são amplamente utilizadas para avaliar tanto jogadores quanto equipes, inclusive estão disponíveis nos próprios sites da ligas. Essas estatísticas normalmente são apresentadas em caráter comparativo, isto é, para classificar jogadores e equipes dentro de um atributo específico. Essa técnica pode ser muito útil para saber o que esperar de um jogador ou time dado seu histórico, porém ela não trás uma visão muito detalhada. Por exemplo, suponha que um jogador faz muitos pontos quando seu time está vencendo, porém quando está atrás no placar, não tem a mesma eficiência. Nesse caso, essa queda de performance recorrente não é retratada numa simples média de pontos por jogo.

Nesse sentido, softwares como o BIRD interpretam esses dados, fornecendo insights e atualizando a equipe técnica em tempo real através de visualizações mais claras e direcionadas para a performance do time e dos atletas em quadra. Porém, ainda não há uma análise voltada especificamente para a performance do jogador de forma que seja possível avaliar sua evolução no tempo e identificar triggers de performance, ou seja, quais condições modificam cada estatística daquele jogador. Essas informações, se bem apresentadas, podem ajudar o técnico escolher os melhores jogadores para colocar em quadra em

um determinado contexto ou a orientar os treinamentos de cada jogador dado a recente evolução de suas estatísticas.

3 Objetivos

Como mencionado no capítulo anterior, é muito importante que a equipe técnica tenha uma visibilidade detalhada das métricas dos jogadores para orientar melhor suas decisões dentro e fora de quadra. Portanto, o propósito desse trabalho é estudar e implementar formas de visualização mais eficientes de estatísticas individuais de jogadores em conjunto com métricas de avaliação de equipes, consolidando as informações mais relevantes no formato de um dashboard no sistema BIRD. A solução será integrada a plataforma BIRD consultando os dados via API e a interface será implementada utilizando a linguagem de programação Angular.

Num primeiro momento, será feito um estudo sobre a API e uma breve análise exploratória para entender as estruturas de dados e rotas disponíveis na API. Além disso, serão pesquisadas as principais formas de avaliação de performance de jogadores de basquete.

Uma vez que a API esteja adequada para realização das análises, será dada continuidade através da elaboração séries temporais das principais métricas de avaliação de performance para representar sua evolução durante uma partida e também durante a temporada, comparando com o histórico anterior.

Em seguida, serão estudadas novas variáveis derivadas dessas métricas que possibilitem uma interpretação mais profunda de possíveis causas para uma determinada variação naquela métrica.

Por fim, será construído o dashboard com as visualizações de cada uma dessas métricas e das variáveis que forem observadas como mais explicativas.

Mesmo tendo sido apresentadas há aproximadamente 20 anos, os Four Factors de Dean Oliver ainda tem uma forte representatividade para times e comissões técnicas ao redor do Mundo. Esses número são responsáveis por medir a capacidade do time de converter arremessos em pontos, proteger a bola durante o ataque, recuperar a bola em arremessos perdidos e converter arremessos da linha de lances livres.

4

Atividades Realizadas

4.1

Estudo sobre os Four Factors de Dean Oliver

No basquete, a performance de uma equipe é influenciada por diversos fatores que afetam diretamente sua eficiência em quadra. Dean Oliver, em seu trabalho sobre estatísticas avançadas de basquete, identificou os Four Factors como os principais determinantes para o sucesso de uma equipe. Esses fatores - Effective Field Goal Percentage (eFG%), Turnover Percentage (TOV%), Rebound Percentage (DRB% e ORB%) e Free Throw Rate (FTR) - fornecem uma visão detalhada sobre a qualidade do ataque, a retenção de posse, o controle de rebotes e a capacidade de gerar pontos em situações de lances livres. Compreender e monitorar esses fatores permite às equipes identificar suas principais forças e fraquezas, ajustando estratégias para otimizar o desempenho e aumentar as chances de vitória em cada partida.

4.1.1

Effective Field Goal Percentage (eFG%)

De acordo com Dean Oliver, a capacidade de converter pontos é um dos principais determinantes para a vitória de uma equipe, sendo representada pelo "Effective Field Goal Percentage" ou eFG%. Segundo Oliver, aproximadamente 40% da performance de um time é influenciada diretamente por esse fator, que pode ser calculado pela fórmula:

$$eFG\% = \frac{\text{arremessos de 2 e 3 convertidos} + 0.5 \times \text{arremessos de 3 convertidos}}{\text{tentativas de arremesso}} \times 100$$

Nesta fórmula, os arremessos de 3 pontos recebem um peso adicional de 0,5, pois um arremesso de 3 pontos vale 50% mais do que um arremesso de 2 pontos. Esse limite é marcado pela linha de 3 pontos, representada na Figura 4.1, que circunda a área conhecida como 'garrafão'. Os limites do garrafão e da linha de 3 pontos podem variar de acordo com as regras de cada confederação organizadora.

Dessa forma, o eFG% oferece uma visão mais completa do aproveitamento de arremessos, considerando a qualidade das oportunidades de pontuação criadas, e equipes com um eFG% elevado tendem a ser mais eficazes no ataque ao maximizar o aproveitamento de suas posses de bola.



Figura 4.1: Linha dos 3 pontos

4.1.2

Turnover Percentage (TOV%)

Um turnover ocorre quando um jogador perde a posse da bola antes de arremessar, seja devido a um passe errado ou a uma interceptação do adversário. Dean Oliver define o percentual de turnover considerando a proporção de posses completas que resultam em turnover, de acordo com a fórmula:

$$TOV\% = \frac{\text{turnovers}}{\text{tentativas de arremesso} + 0.44 \times \text{tentativas lance livre} + \text{turnovers}} \times 100$$

Para calcular corretamente o número de posses completas, Oliver introduz um fator de ajuste estatístico importante, o fator de 0,44 aplicado às tentativas de lance livre. Esse ajuste se dá porque nem todas as tentativas de lance livre encerram uma posse. Em situações de lance livre, uma única tentativa pode não representar a conclusão da posse, pois pode haver uma sequência de lances livres ou um rebote ofensivo após um lance perdido, que prolonga a posse. Assim, o fator 0,44 foi obtido empiricamente para refletir aproximadamente o impacto médio das tentativas de lance livre no número de posses completas.

Portanto, o TOV% oferece uma métrica confiável para avaliar a eficiência da equipe na retenção da posse de bola em relação ao total de posses. Um baixo TOV% indica que o time comete poucos turnovers, o que contribui para uma melhor manutenção da posse e, conseqüentemente, mais oportunidades de finalização. Para Oliver, essa métrica é responsável por explicar aproximadamente 25% da performance da equipe.

4.1.3

Rebound Percentage (DRB% e ORB%)

As porcentagens de rebotes defensivos (DRB%) e ofensivos (ORB%) são métricas essenciais no basquete para avaliar o controle de rebotes de uma equipe ou jogador. Estas estatísticas ajudam a determinar o domínio de uma equipe na recuperação de posse de bola, seja ao impedir que o adversário tenha uma segunda chance de pontuar (rebote defensivo) ou ao gerar uma nova oportunidade de ataque para a própria equipe (rebote ofensivo). Segundo Dean Oliver, o percentual de rebotes representa cerca de 25% da performance da equipe.

4.1.3.1

Defensive Rebound Percentage (DRB%)

A Defensive Rebound Percentage (DRB%) mede a proporção de rebotes defensivos capturados por um jogador ou equipe em relação ao total de rebotes disponíveis no lado defensivo da quadra. Essa métrica é fundamental para impedir que a equipe adversária continue na posse de bola após um arremesso não convertido, limitando suas chances de pontuação extra. A fórmula para o DRB% é dada por:

$$DRB\% = \frac{\text{rebotes defensivos}}{\text{rebotes ofensivos do adversário} + \text{rebotes defensivos}} \times 100$$

Um DRB% elevado indica eficiência na recuperação de rebotes defensivos, o que contribui para uma transição rápida ao ataque e reduz as oportunidades de segunda chance do adversário. Equipes com altos valores de DRB% geralmente apresentam uma defesa mais sólida, pois controlam melhor o espaço defensivo e limitam as chances de pontuação do oponente.

4.1.3.2

Offensive Rebound Percentage (ORB%)

A Offensive Rebound Percentage (ORB%) representa a proporção de rebotes ofensivos capturados por um jogador ou equipe em relação ao total de rebotes disponíveis no lado ofensivo da quadra. Essa métrica reflete a capacidade de uma equipe de gerar novas posses de bola após tentativas de arremesso falhas, criando mais oportunidades de pontuar. A fórmula para o ORB

$$ORB\% = \frac{\text{rebotes ofensivos}}{\text{rebotes defensivos do adversário} + \text{rebotes ofensivos}} \times 100$$

Um ORB% alto indica que a equipe é eficiente em buscar rebotes ofensivos, o que aumenta o número de tentativas de arremesso e, potencialmente, o total de pontos. Times com valores elevados de ORB% podem manter a pressão ofensiva sobre o adversário e aproveitar ao máximo suas posses de bola, compensando, por vezes, uma baixa eficiência nos arremessos.

4.1.4

Free Throw Rate (FTR)

O Free Throw Rate (FTR), ou Taxa de Lances Livres, é uma métrica utilizada para avaliar a frequência com que uma equipe ou jogador consegue converter posses em tentativas de lance livre, a partir da proporção entre o número de lances livres e arremessos. A fórmula para calcular o FTR é dada por:

$$FTR\% = \frac{\text{tentativas de lance livre}}{\text{tentativas de arremesso}} \times 100$$

Um FTR elevado sugere que a equipe é eficiente em provocar faltas do adversário, o que traz diversos benefícios. Primeiramente, os lances livres oferecem uma oportunidade de pontuar sem a pressão da defesa adversária. Além disso, ao acumular faltas, a equipe adversária pode ser forçada a reduzir a agressividade defensiva ou a tirar jogadores importantes de quadra para evitar que atinjam o limite de faltas. Essa métrica representa, segundo Dean Oliver, 15% da performance da equipe.

4.2

Estudo sobre a API BIRD

A API BIRD, criada pelo Instituto Tecgraf da PUC-Rio em parceria com o Flamengo, facilita o acesso a dados analíticos sobre partidas, jogadores e times, com base em informações pela NBB. Ela disponibiliza endpoints que permitem obter dados detalhados de eventos ocorridos em jogo, de forma acessível e eficiente para análises técnicas e estratégicas. Essa API visa simplificar o uso desses dados para visualizações e estudos de performance no contexto esportivo. Todas as rotas disponíveis estão especificadas na documentação da API.

4.2.1

Principais Endpoints

As principais rotas da API para o desenvolvimento de análises estatísticas dos eventos são as rotas de ligas, times, jogadores, partidas e eventos, pois a partir delas é possível selecionar um contexto específico - como uma liga, uma temporada, um time ou partida - e calcular os indicadores desejados a partir dos eventos relacionados a esse contexto.

4.2.1.1

GET /v1/leagues/

As rotas de ligas, mostradas na Figura 4.2, permitem acessar informações das ligas disponíveis e suas temporadas associadas. Esse conjunto de rotas é útil para identificar competições específicas e comparar desempenho entre diferentes períodos.

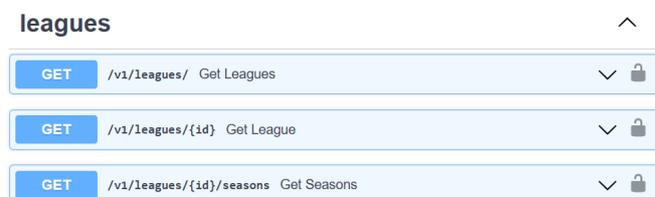


Figura 4.2: Rotas de Ligas

4.2.1.2

GET /v1/teams/

As rotas de times, ilustradas na Figura 4.3, fornecem dados sobre times específicos, incluindo acrônimos, nomes completos e detalhes de temporada. Essas rotas também possibilitam obter informações detalhadas de um time a partir do seu id, de modo que não seja possível consulta-las de maneira pontual, por exemplo quando deseja-se obter os acrônimos dos times participantes de uma determinada partida.

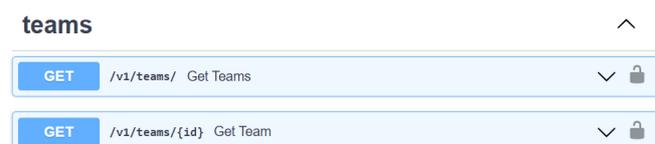
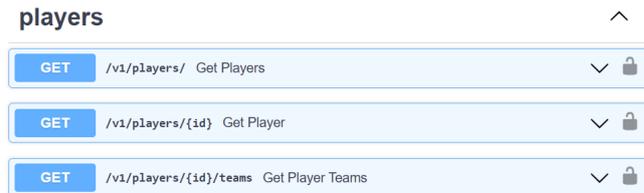


Figura 4.3: Rotas de Times

4.2.1.3

GET /v1/players/

As rotas de jogadores, apresentadas na Figura 4.4, são projetadas para fornecer detalhes de cada jogador, identificados por um ID único. Esse endpoint permite tanto listar os jogadores de um time quanto obter informações detalhadas sobre um jogador específico, facilitando o rastreamento de sua performance ao longo de eventos.



The screenshot shows a list of API routes under the heading 'players'. Each route is displayed in a light blue box with a 'GET' method indicator on the left, the endpoint path in the middle, and the description on the right. A dropdown arrow and a lock icon are visible on the right side of each entry.

Method	Endpoint	Description
GET	/v1/players/	Get Players
GET	/v1/players/{id}	Get Player
GET	/v1/players/{id}/teams	Get Player Teams

Figura 4.4: Rotas de Jogadores

4.2.1.4

GET /v1/matches/

As rotas de partidas, ilustradas na Figura 4.5, são essenciais para acessar dados de jogos específicos. Elas permitem verificar se uma partida está em andamento, visualizar o placar, ou acessar uma lista de partidas de uma temporada, facilitando o acompanhamento do desempenho ao longo de competições.



The screenshot shows a list of API routes under the heading 'matches'. Each route is displayed in a light blue box with a 'GET' method indicator on the left, the endpoint path in the middle, and the description on the right. A dropdown arrow and a lock icon are visible on the right side of each entry.

Method	Endpoint	Description
GET	/v1/matches/{league_id}	Get League Matches
GET	/v1/matches/{league_id}/{match_id}	Get Match
GET	/v1/matches/{league_id}/{match_id}/score	Get Match Score
GET	/v1/matches/{league_id}/{match_id}/ongoing	Is Match Ongoing
GET	/v1/matches/{league_id}/{match_id}/quarters	Get Match Quarters
GET	/v1/matches/{league_id}/{match_id}/players	Get Match Players
GET	/v1/matches/{league_id}/{match_id}/teams	Get Match Teams
GET	/v1/matches/{league_id}/{match_id}/runs	Get Match Runs
GET	/v1/matches/{league_id}/{match_id}/playtimes	Get Match Playtimes

Figura 4.5: Rotas de Partidas

4.2.1.5

GET /v1/events/

As rotas de eventos, mostradas na Figura 4.6, são fundamentais para cálculos estatísticos, pois registram eventos de jogo específicos, como arremessos e faltas. Um dos endpoints também permite mapear o significado dos códigos de evento, facilitando a interpretação dos dados pelos desenvolvedores.



The image shows a list of API endpoints under the heading 'events'. Each endpoint is represented by a light blue box containing the HTTP method (GET), the URL path, a description, and a lock icon. The endpoints are:

Method	Endpoint	Description
GET	/v1/events/types	Get Event Types
GET	/v1/events/{league_id}/{match_id}	Get Match Events
GET	/v1/events/favorites	Get Favorite Events
GET	/v1/events/playlist/{playlist_id}	Get Playlist Events
GET	/v1/events/{league_id}/{match_id}/{event_id}	Get Event
GET	/v1/events/{league_id}/{match_id}/{event_id}/video	Get Event Video

Figura 4.6: Rotas de Eventos

4.3

Protótipos de Gráficos em Python

Nesta etapa do projeto, com o estudo da API concluído, o objetivo era desenvolver protótipos de gráficos que representassem a evolução dos dados de um jogador em uma partida, processando as informações retornadas pela API. A Figura 4.7 mostra um exemplo desses protótipos, onde a linha azul ilustra a progressão da pontuação de um jogador ao longo do tempo, com indicadores visuais para entradas e saídas de quadra (respectivamente, pontos verdes e vermelhos). Para construir esse tipo de visualização, foi necessário lidar com as limitações da API, que retorna os eventos de uma partida de forma paginada.

Esse fator exigiu a implementação de um mecanismo eficiente para recuperar e organizar os eventos de uma partida inteira, especialmente pensando em jogos que ainda estivessem em andamento. Nesses casos, seria necessário fazer requisições periódicas para verificar novos dados e atualizar a visualização em tempo real. Assim, foi essencial planejar uma estratégia de consulta que otimizasse as requisições, evitando sobrecarregar o sistema com dados redundantes, enquanto permitia uma atualização precisa e contínua dos gráficos para os usuários.

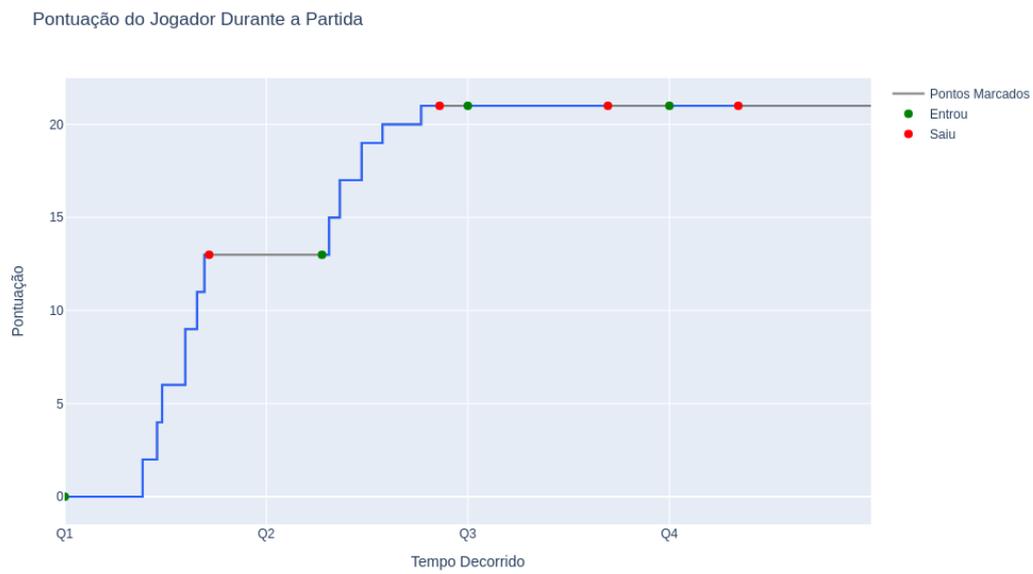


Figura 4.7: Pontuação de um jogador durante a partida

5

Projeto e Especificação do Sistema

5.1

Visão Geral do Sistema

O sistema desenvolvido tomou como ponto de partida o software BAICA, que já era utilizado pela comissão técnica do Flamengo. No entanto, com a transição para uma interface web através do BIRD, o BAICA deixou de receber atualizações e novas funcionalidades. Assim, tornou-se necessário migrar algumas telas exclusivas do BAICA para o BIRD.

Além disso, como discutido nos capítulos anteriores, a implementação dos “Four Factors” no BAICA apresentava limitações, pois não permitia analisar a contribuição individual de cada jogador para esses fatores. Embora os “Four Factors” tenham sido originalmente criados para avaliar o desempenho de equipes, alguns desses fatores são cada vez mais aplicados na avaliação individual de jogadores. O novo sistema foi projetado para superar essa limitação, oferecendo uma interface de visualização que facilita a interpretação das contribuições de cada atleta de forma mais clara e detalhada. Dessa forma, o sistema permite à comissão técnica um entendimento mais preciso do impacto individual dos jogadores em cada um dos fatores de desempenho.

As Figuras 5.1 e 5.2 mostram um comparativo entre as duas interfaces, destacando as melhorias implementadas no BIRD.

5.2

Arquitetura do Sistema

A interface do sistema foi desenvolvida utilizando o framework Angular. O Angular é uma plataforma robusta e versátil, criada pelo Google, que permite o desenvolvimento de aplicações de página única (Single-Page Applications - SPAs) de forma eficiente e organizada. Esse framework combina uma estrutura de design de aplicações com um conjunto de ferramentas para facilitar o desenvolvimento de interfaces sofisticadas e interativas.

O Angular oferece diversos recursos que contribuem para a criação de aplicações modernas, como a organização modular do código, que facilita a manutenção e escalabilidade do projeto, e o uso de componentes reutilizáveis, que permitem a criação de interfaces dinâmicas. Embora esses recursos tenham sido utilizados de forma limitada neste projeto, o Angular proporcionou a

Four Factors

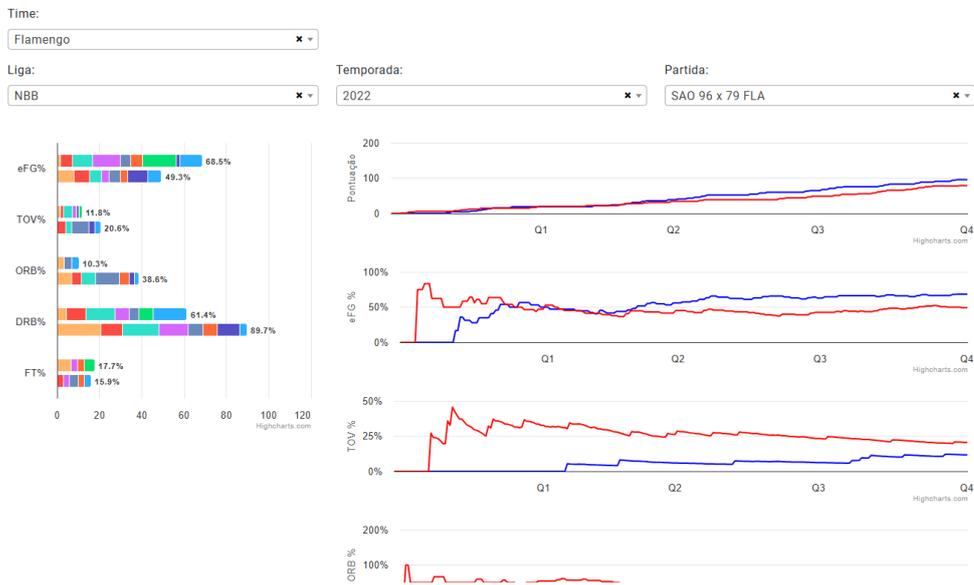


Figura 5.1: Interface desenvolvida para o BIRD



Figura 5.2: Interface BAICA

possibilidade de desenvolver a interface de forma independente da aplicação oficial do BIRD, permitindo a fácil inclusão dos componentes criados.

Além disso, o Angular conta com um sistema de injeção de dependência, que facilita a integração entre os diferentes módulos da aplicação, e um poderoso sistema de data binding, que garante que as mudanças nos dados sejam refletidas automaticamente na interface do usuário. Isso foi particularmente útil durante o desenvolvimento, pois qualquer alteração feita no código era refletida imediatamente na interface enquanto a aplicação estava em execução, diferentemente de outras soluções que poderiam exigir uma recompilação ou a criação de uma nova versão para aplicar as mudanças.

O Angular utiliza como linguagem principal o TypeScript, uma extensão do JavaScript que adiciona tipagem estática e outros recursos avançados ao código. O TypeScript oferece maior segurança e facilita a manutenção do projeto, pois ajuda a identificar erros de tipagem durante o desenvolvimento, antes mesmo de o código ser executado. Essa tipagem estática é particularmente útil em projetos de grande escala, pois torna o código mais previsível e fácil de refatorar.

Por outro lado, essa tipagem rigorosa também trouxe alguns desafios ao desenvolvimento, especialmente ao lidar com objetos cujas chaves são variáveis. Nesses casos, o TypeScript exige que a estrutura dos dados seja definida de maneira precisa, o que pode dificultar o acesso a valores específicos em objetos dinâmicos. Para contornar essa limitação, foi necessário definir interfaces de dados para representar a estrutura desses objetos, permitindo que o TypeScript reconhecesse as propriedades e validasse o acesso a seus valores. Essa abordagem garantiu maior consistência no código, ainda que tenha demandado um esforço adicional para especificar essas interfaces.

Código 1: Interfaces Criadas

```
1 interface FourFactorsMetrics {
2     efg: number | null;
3     tov: number | null;
4     orb: number | null;
5     drb: number | null;
6     ft: number | null;
7 }
8
9 interface FourFactorsData {
10    elapsed_time: number;
11    home: FourFactorsMetrics;
12    away: FourFactorsMetrics;
13 }
```

Além do Angular e do TypeScript, foram utilizadas outras bibliotecas e frameworks no front-end para tornar a interface mais rica e interativa. Uma dessas bibliotecas é o Bootstrap, um popular framework CSS que facilita a criação de componentes visuais responsivos e adaptáveis a diferentes tamanhos de tela. No projeto, o Bootstrap foi utilizado para criar e estilizar elementos como selects e modais, garantindo que a interface fosse intuitiva e visualmente agradável. Um exemplo de uso do Bootstrap está na inicialização dos selects com o componente `select2`, que fornece uma experiência de seleção aprimorada e estilizada, como mostrado na Figura 5.3.

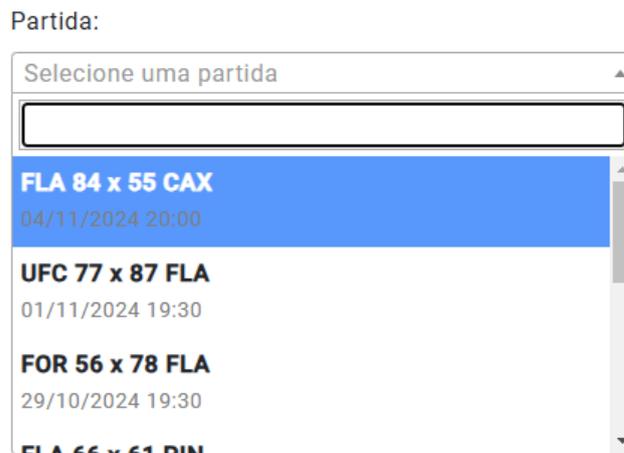


Figura 5.3: Select Estilizado com Bootstrap

Para a criação dos gráficos, foi utilizada a biblioteca Highcharts, uma poderosa ferramenta de visualização de dados em JavaScript, gratuita para fins pessoais e não comerciais. O Highcharts permite a criação de gráficos interativos e altamente customizáveis, oferecendo suporte a diversos tipos de gráficos, como linhas, barras, colunas e outros. No projeto, o Highcharts foi essencial para visualizar os dados dos Four Factors de forma clara e interativa, permitindo à equipe técnica uma análise visual intuitiva e detalhada das métricas, com um alto grau de personalização.

O uso dessa biblioteca também facilitou a implementação de recursos adicionais envolvendo eventos, como a sincronização entre gráficos quando um período específico é selecionado para recalcular os fatores dentro daquele intervalo. Esse recurso, ilustrado nas Figuras 5.4 e 5.5, permite que a equipe técnica avalie a performance do time em momentos específicos, considerando apenas os eventos ocorridos dentro do intervalo selecionado.

Além disso, foi utilizado o jQuery, uma biblioteca JavaScript que simplifica o acesso e a manipulação de elementos HTML na página. O jQuery

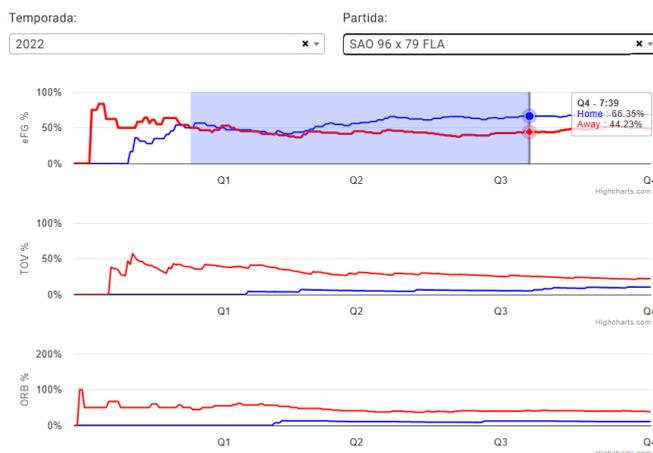


Figura 5.4: Selecionando Um Intervalo

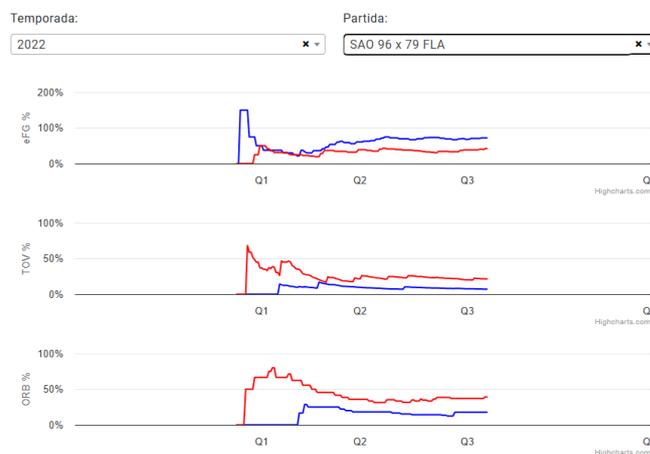


Figura 5.5: Aplicação Sincronizada do Intervalo

é conhecido por seu seletor simplificado e por métodos que facilitam a manipulação do DOM (Document Object Model), além de facilitar a realização de animações e o gerenciamento de eventos. No projeto, o jQuery foi particularmente útil para simplificar a comunicação com a API através de métodos AJAX, permitindo que o front-end fizesse requisições assíncronas ao back-end para buscar e atualizar dados sem recarregar a página. Esse recurso foi essencial para melhorar a experiência do usuário, mantendo a aplicação responsiva e rápida, uma vez que é possível especificar quais ações irão ocorrer antes do envio da requisição e depois, tanto no caso de sucesso quanto de falha.

5.3

Processamento de Dados

O processamento dos dados ocorre da seguinte forma: ao selecionar uma partida, é realizada uma requisição assíncrona para obter a primeira página de

eventos dessa partida. Cada página retornada pela API contém no máximo 100 eventos, que são armazenados em um array de eventos. Se a quantidade total de eventos da partida, também fornecida pela API, for maior que o tamanho do array atual, uma nova requisição é feita para carregar a próxima página. Esse processo é repetido até que todas as páginas de eventos da partida tenham sido retornadas.

Com todas as páginas carregadas, o array de eventos é processado iterativamente. Para cada evento, as estatísticas necessárias são calculadas, e os Four Factors são atualizados para refletir cada momento em que ocorreu um evento.

Como a interface será utilizada também durante o andamento das partidas, o número da última página de eventos carregada é armazenado. Assim, a aplicação realiza periodicamente uma nova requisição a essa página para verificar a ocorrência de novos eventos. Caso novos eventos sejam detectados, os fatores são calculados e adicionados ao array que armazena os valores dos fatores ao longo do tempo. Esse processo evita o recálculo dos fatores retroativamente. Se a nova página de eventos atingir o limite de 100 eventos, o número da página é atualizado para a próxima, permitindo que a aplicação consulte sempre a página correta para obter novos dados.

O processo completo de carregamento, processamento e atualização dos dados de eventos é ilustrado no Algoritmo 1. Esse algoritmo descreve o fluxo necessário para garantir que os dados estejam sempre atualizados durante o andamento da partida, maximizando a eficiência ao evitar o recálculo desnecessário dos fatores.

Algoritmo 1: Processamento de dados e cálculo dos Four Factors

Entrada: partida selecionada**Saída:** array de Four Factors por Momento

```
1 pagina_atual ← 1
2 eventos_total ← obterTotalEventos(partida_selecionada)
3 array_eventos ← vazio
4 enquanto tamanho(array_eventos) < eventos_total faça
5   | eventos ← requisitarPaginaEventos(partida_selecionada,
6   |   pagina_atual)
7   | array_eventos.adicionar(eventos)
8   | se tamanho(eventos) == 100 então
9   |   | pagina_atual ← pagina_atual + 1
10  |   | senão
11  |   |   | parar
11 para cada evento em array_eventos faça
12   | atualizarEstatisticas(evento)
13   | calcularFourFactors(evento)
```

6

Implementação e Avaliação

Para verificar o correto funcionamento das funcionalidades implementadas, foram realizados três principais testes: um para testar a usabilidade da interface, outro para validar as fórmulas utilizadas nos cálculos, e um terceiro para avaliar o desempenho da interface em tempo real.

6.1

Validação das Funcionalidades da Interface

Este primeiro teste teve como objetivo avaliar as funcionalidades da interface, como o funcionamento dos selects, o preenchimento correto dos gráficos e as ações interativas implementadas. Abaixo, detalhamos cada aspecto da interface testado e os ajustes realizados.

6.1.1

Seleção de Jogos e Times

Inicialmente, a interface foi desenvolvida para exibir apenas jogos do Flamengo, sem a opção de selecionar outros times. Nesse contexto, não era necessário um select para escolher o time, e a lista de jogos era filtrada para mostrar apenas partidas do Flamengo. No entanto, durante a fase de testes, identificou-se a necessidade de permitir a visualização de jogos de outros times, a fim de possibilitar comparações e análises mais amplas. Assim, o filtro que restringia os jogos ao Flamengo foi removido, e um select foi adicionado para permitir a seleção de qualquer time de referência.

Durante os testes dos selects, notou-se que o número total de jogos na temporada para todos os times ultrapassava o limite de 100 itens por página definido pela API, tornando as requisições paginadas. Para contornar esse problema, a seleção de um time específico foi mantida, garantindo que o número de jogos exibidos estivesse sempre dentro do limite da API.

6.1.2

Atualização dos Gráficos com Eventos da Partida

Nos testes relacionados aos gráficos, o primeiro aspecto avaliado foi se os dados estavam sendo atualizados corretamente com base nos eventos ocorridos durante as partidas. Foram analisadas diversas partidas, registrando-se os eventos de cada uma para verificar se os cálculos dos Four Factors refletiam com precisão esses eventos.

Durante essa etapa, foi identificado um problema com eventos que ocorriam simultaneamente no relógio da partida, como uma falta seguida de um lance livre enquanto o relógio está parado. Esse cenário gerava duplicidade de pontos no gráfico, resultando em múltiplos valores no eixo y (Four Factors) para o mesmo ponto no eixo x (tempo da partida). Para resolver essa questão, foi implementada uma verificação que descarta o valor calculado na última atualização se o tempo do evento atual for igual ao do último valor salvo.

6.1.3 Funcionalidades Interativas dos Gráficos

Foram também testadas as funcionalidades interativas dos gráficos, que incluíram:

- **Seleção de Intervalo:** permite recalculer os indicadores para um intervalo específico da partida.
- **Seleção de Ponto Específico:** possibilita a visualização de um ponto específico no gráfico de barras.
- **Abertura do Modal de Legenda Detalhada:** exibe uma legenda detalhada para facilitar a compreensão dos dados apresentados nos gráficos de barras, conforme a figura 6.1.



Figura 6.1: Modal de Legenda

Durante esses testes, foram identificados dois problemas. O primeiro estava relacionado a um desalinhamento entre os eixos dos gráficos, causado pela variação na largura dos rótulos do eixo y. Dependendo do formato dos valores (número de dígitos e presença ou ausência de porcentagem), o espaço ocupado pelo eixo variava, gerando um desalinhamento no eixo x entre os gráficos, conforme mostrado na Figura 6.2. Para solucionar esse problema, foi

implementada uma barra de referência de tempo entre os gráficos, que indica o ponto selecionado em todos os gráficos, facilitando a comparação visual, como ilustrado na Figura 6.3.

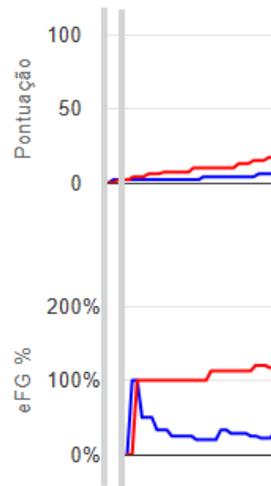


Figura 6.2: Desalinhamento entre os gráficos devido aos rótulos do eixo y

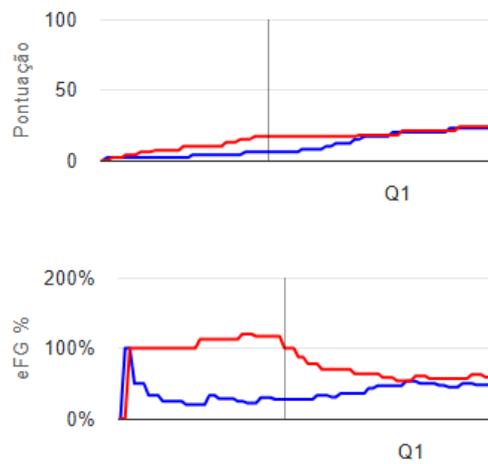


Figura 6.3: Barra de referência de tempo entre os gráficos

O segundo problema foi encontrado na funcionalidade de seleção de intervalos. Houve um erro na função que recalculava os Four Factors para o intervalo selecionado. Essa função inseria valores nulos para todos os eventos fora do intervalo, mas não tratava a duplicidade de pontos mencionada anteriormente, o que causava desalinhamento no eixo x. A função foi então ajustada para remover os pontos duplicados ao recalculer os fatores dentro do intervalo.

6.1.4

Validação do Modal de Legenda

A última funcionalidade testada foi a abertura do modal de detalhamento da legenda do gráfico de barras. Não foram encontrados erros ou problemas de implementação, de modo que os valores exibidos no modal eram consistentes com os valores apresentados na tela principal, indicando que a funcionalidade estava correta.

6.2

Validação das Fórmulas

O segundo teste focou na validação das fórmulas aplicadas para o cálculo dos Four Factors, verificando se os eventos estavam sendo corretamente identificados e processados e se as métricas eram calculadas conforme esperado. Para realizar essa validação, foi solicitado à comissão técnica do Flamengo um print da tela de Four Factors de um jogo específico no sistema BAICA, conforme mostrado na Figura 5.2. Com base nesse print, foi possível localizar o jogo correspondente na API e comparar os valores dos Four Factors calculados no sistema com os valores exibidos no BAICA.

Durante esse teste, constatou-se que todos os fatores estavam sendo calculados corretamente, com exceção do TOV% (Turnover Percentage). Ao buscar esclarecimentos com a equipe técnica do Flamengo, foi informado que eles haviam modificado a fórmula original de Dean Oliver para atender a uma necessidade específica do treinador. A fórmula adaptada fornecida pela equipe é descrita abaixo:

$$TOV\% = \frac{\text{Turnovers}}{\text{Posses do Jogo}} \times 100$$

onde:

$$\text{Posses do Jogo} = \frac{\text{Posses do Time} + \text{Posses do Adversário}}{2}$$

A posse de um time é calculada pela fórmula:

$$\begin{aligned} \text{Posses} = & \text{Tentativas de Arremesso} + 0.4 \times \text{Tentativas de Lance Livre} \\ & - 1.07 \times \frac{\text{ORB}\%}{100} \times \text{Arremessos não Convertidos} + \text{Turnovers} \end{aligned}$$

Após a realização do ajuste necessário para incorporar a fórmula adaptada, verificou-se que todos os valores calculados correspondiam aos valores

esperados. Assim, a validação das fórmulas dos “Four Factors” foi concluída com sucesso, garantindo que o sistema estivesse em conformidade com as métricas específicas utilizadas pela comissão técnica.

6.3

Testes em Tempo Real

O objetivo desta etapa foi avaliar o funcionamento das funcionalidades para partidas em andamento, com ênfase na atualização periódica de eventos e na atualização correta dos gráficos. Esse teste verificou se os eventos estavam sendo obtidos e processados corretamente em intervalos regulares e se os gráficos refletiam essas atualizações em tempo real.

O gráfico principal analisado foi o de pontuação. A pontuação foi escolhida como referência para a validação, pois, considerando que os cálculos dos Four Factors já haviam sido previamente testados, uma atualização correta da pontuação indicaria que os eventos estavam sendo processados adequadamente. Assim, conclui-se que, se a pontuação estivesse precisa, os fatores calculados em tempo real também estariam corretos. Como é possível observar nas Figuras 6.4 e 6.5 o placar ao final do primeiro quarto estava correto.



Figura 6.4: Transmissão da partida entre Brasília e Unifacisa ao Vivo

Esse método de validação foi adotado devido à impossibilidade de realizar um teste paralelo com o sistema BAICA para um jogo em andamento, o que limitou as comparações diretas entre os sistemas.

Four Factors



Figura 6.5: Interface durante partida entre Brasília e Unifacisa ao Vivo

7

Consideração Finais

Este capítulo encerra a análise e reflexão sobre o desenvolvimento e os resultados obtidos no projeto. Aqui, são discutidas as contribuições realizadas, as lições aprendidas, possíveis melhorias e sugestões para trabalhos futuros.

7.1

Contribuições do Trabalho

O principal objetivo deste projeto foi desenvolver uma ferramenta que permitisse uma análise aprofundada dos “Four Factors” e fosse integrada ao sistema Bird, de forma semelhante ao modelo implementado anteriormente no BAICA. Este objetivo foi alcançado com sucesso, resultando na recriação da interface de visualização dos “Four Factors” e na sua adaptação para o Bird, atendendo a uma demanda específica da comissão técnica do Flamengo.

Além de replicar a visualização, o trabalho agregou valor à análise esportiva ao introduzir uma métrica que mensura a contribuição individual de cada jogador em relação a cada fator. Isso, somado a componentes gráficos adicionais e uma interface visual aprimorada, expandiu as possibilidades de análise ao permitir o cálculo dos fatores para momentos e intervalos específicos, de forma assíncrona. Dessa forma, a ferramenta oferece maior flexibilidade para análises detalhadas e customizadas, potencializando a capacidade da comissão técnica em tomar decisões informadas.

7.2

Aprendizado com o Trabalho

Este projeto proporcionou uma série de aprendizados práticos e teóricos. Do ponto de vista técnico, a implementação exigiu o domínio de ferramentas de análise de dados, incluindo bibliotecas específicas para visualização de dados, além de um aprofundamento no uso de TypeScript e Angular. Aprender a trabalhar com TypeScript foi especialmente relevante, tanto pela necessidade de dominar suas particularidades em relação ao JavaScript puro quanto pelo seu papel na estruturação e segurança do código. O aprendizado sobre o framework Angular, com seu modelo baseado em componentes e uso de templates dinâmicos, foi fundamental para criar uma interface robusta e responsiva, tornando a etapa prototipagem mais simples.

Metodologicamente, o trabalho reforçou a importância de um planejamento estruturado e da aplicação de uma abordagem de desenvolvimento in-

cremental, que permitiu adaptações e refinamentos contínuos ao longo do processo. Foi enriquecedor também explorar conceitos avançados de análise estatística aplicada ao esporte, uma área em rápido crescimento que vem transformando o esporte moderno. Esses conhecimentos não só aprimoraram a análise técnica do projeto, como também ampliaram a compreensão sobre a relevância da análise de dados no contexto esportivo atual.

7.3

Limitações e Oportunidades de Melhoria

Ao avaliar o projeto como um todo, foram identificadas algumas limitações e oportunidades de melhoria. A primeira limitação está relacionada à ausência de uma funcionalidade de alarmes. Embora essa funcionalidade não fosse um foco central neste projeto, seria possível implementar uma interface de alertas semelhante à existente no BAICA, na qual o treinador ou a equipe técnica poderiam configurar notificações para quando um fator atingisse um nível específico durante a partida, auxiliando em decisões táticas em tempo real.

Outra oportunidade de aprimoramento seria incluir uma visualização temporal dos fatores para jogadores individuais, similar à visualização atualmente disponível para as equipes. Esse recurso permitiria um monitoramento mais detalhado do desempenho individual ao longo da partida. No entanto, é importante considerar uma forma de visualização que evite a sobrecarga de informações, pois ao exibir todos os jogadores simultaneamente poderia resultar em uma interface visualmente poluída. Uma solução seria implementar uma tela de comparação, na qual fosse possível selecionar e comparar dois jogadores, visualizando seus fatores ao longo da partida de forma clara e organizada.

7.4

Recomendações para Trabalhos Futuros

Para desenvolvimentos futuros, a incorporação de técnicas de machine learning representa uma oportunidade promissora para ampliar as funcionalidades e o valor analítico da ferramenta. Utilizando dados históricos dos Four Factors, seria possível criar modelos preditivos que fornecessem insights sobre o desempenho futuro de equipes e jogadores. Esses modelos poderiam identificar padrões e tendências de desempenho, oferecendo previsões que auxiliem a comissão técnica a tomar decisões estratégicas mais embasadas.

Algumas abordagens que poderiam ser exploradas incluem algoritmos de aprendizado supervisionado para previsão de desempenho individual e coletivo, análise de regressão para identificar fatores mais influentes sobre os resultados,

e clustering para segmentar jogadores ou equipes em perfis de desempenho semelhantes. Modelos de séries temporais também poderiam ser aplicados para capturar a evolução dos “Four Factors” ao longo das temporadas, permitindo antecipar tendências e ajustar estratégias conforme o desenvolvimento da equipe.

Além disso, seria útil implementar sistemas de recomendação baseados em machine learning, que sugerissem ajustes táticos durante uma partida com base no comportamento e no impacto dos Four Factors em tempo real. Essa abordagem integraria a análise preditiva com a geração de recomendações acionáveis, transformando o sistema em uma ferramenta proativa e ainda mais eficaz no suporte à tomada de decisões esportivas.

8

Referências bibliográficas

BASKETBALL.ORG. **Site basketball.org**. 2024. Disponível em: <<https://www.basketball.org/3-point-shot-distances/>>.

FIBA. **Basketball Terminology**. 2023. Disponível em: <<https://www.fiba.basketball/europe/FECC/FIBA-Europe-Basketball-Terminology-2023.pdf>>.

FLAMENGO, C. de R. **Flamengo e PUC-Rio fecham parceria de tecnologia e inovação**. 2020. Disponível em: <<https://www.flamengo.com.br/noticias/institucional/flamengo-e-puc-rio-fecham-parceria-de-tecnologia-e-inovacao>>.

FORBES. **Moneyball 20 Years Later: A Progress Report On Data And Analytics In Professional Sports**. 2022. Disponível em: <<https://www.forbes.com/sites/randybean/2022/09/18/moneyball-20-years-later-a-progress-report-on-data-and-analytics-in-professional-sports/?sh=65987b05773d>>.

GOOGLE. **Documentação Angular**. 2024. Disponível em: <<https://v17.angular.io/docs>>.

NBA. **Site oficial NBA**. 2024. Disponível em: <<https://www.nba.com/>>.

NBB. **Site oficial NBB**. 2024. Disponível em: <<https://lnb.com.br/nbb/>>.

ORACLE. **What is an API?** 2024. Disponível em: <<https://www.oracle.com/br/cloud/cloud-native/api-management/what-is-api/>>.

TECGRAF, I. **Documentação BIRD API**. 2024. Disponível em: <<https://apibird.tecgraf.puc-rio.br/docs>>.

VEJA, R. **Viva o basquete no país do futebol**. 2023. Disponível em: <<https://veja.abril.com.br/esporte/viva-o-basquete-no-pais-do-futebol>>.