

**Instituto Politécnico do Cávado e do Ave**

*Licenciatura em Engenharia de Sistemas Inforáticos*

*Integração de Sistemas de Informação*

---

---

**Trabalho Prático 1**

---

Estudante: Rui Rodrigues

Professor: Luís Ferreira

22 de outubro de 2024

## Índice

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Problema</b>	<b>5</b>
<b>3</b>	<b>Estratégia Utilizada</b>	<b>5</b>
<b>4</b>	<b>Transformações</b>	<b>5</b>
<b>5</b>	<b>Diagrama da Padronização e Limpeza</b>	<b>6</b>
5.1	Seleção e Padronização de Campos . . . . .	6
5.2	Verificação de Nomes . . . . .	7
<b>6</b>	<b>Diagrama de Filtragem e Cálculo de Performance</b>	<b>9</b>
6.1	Filtragem . . . . .	9
6.2	Cálculo de Performance . . . . .	10
<b>7</b>	<b>Diagrama de Conexão com Base de Dados</b>	<b>11</b>
7.1	Substituição de Strings . . . . .	11
<b>8</b>	<b>Diagrama dos 5 Melhores Jogadores por Posição</b>	<b>12</b>
8.1	Agrupamento por Posição . . . . .	12
8.2	Ordenação dos Dados . . . . .	13
8.3	Seleção de Dados . . . . .	13
<b>9</b>	<b>Job</b>	<b>15</b>
<b>10</b>	<b>Node-RED</b>	<b>16</b>
<b>11</b>	<b>Vídeo com demonstração (QR Code)</b>	<b>18</b>
<b>12</b>	<b>Conclusão</b>	<b>19</b>
<b>13</b>	<b>Bibliografia</b>	<b>20</b>

## Lista de Figuras

1	Fluxo da Primeira Transformação . . . . .	6
2	Seleção de Campos . . . . .	7
3	Tratamento de Valores Nulos . . . . .	7
4	Verificação dos Nomes . . . . .	8
5	Filtragem e Cálculo de Performance . . . . .	9
6	Filtragem por Número de Jogos . . . . .	10
7	Cálculo Performance Score . . . . .	10
8	Conexão com Base de Dados . . . . .	11
9	Substituição de Vírgulas por Pontos . . . . .	11
10	Seleção dos 5 Melhores Jogadores por Posição . . . . .	12
11	Agrupamento por Posição . . . . .	13
12	Ordenação por Performance . . . . .	13
13	Seleção 5 melhores . . . . .	14
14	Job . . . . .	15
15	Node-Red . . . . .	16
16	Gráfico . . . . .	16
17	Jogadores Chelsea . . . . .	17
18	QR-code . . . . .	18

# 1 Introdução

Este trabalho tem como objetivo a análise de um conjunto de dados de jogadores da Premier League. A partir desses dados, serão realizadas transformações e agrupamentos para identificar padrões de desempenho dos jogadores e selecionar os 5 melhores em cada posição.

O conjunto de dados está no formato CSV e contém as estatísticas dos jogadores referentes ao ano de 2020.

## 2 Problema

O principal objetivo é realizar uma análise detalhada dos dados estatísticos dos jogadores da Premier League, com o intuito de identificar os cinco melhores jogadores em cada posição.

## 3 Estratégia Utilizada

Com base no problema proposto, a estratégia de abordagem consistiu nas seguintes etapas:

- **Normalização e Verificação dos Dados dos Jogadores:** Padronização dos dados, assegurando que campos como nomes, clubes e posições estejam uniformes em termos de formatação.
- **Agrupamento dos Dados:** Organização dos dados de acordo com a posição dos jogadores (guarda-redes, defesas, médios e atacantes) para possibilitar a análise de métricas específicas relacionadas às funções de cada jogador.
- **Determinação do Desempenho por Posição:** Definição de métricas-chave que indicam o desempenho de jogadores em cada posição.

## 4 Transformações

Foram realizadas quatro transformações principais: padronização e normalização dos dados, cálculos para identificar os melhores jogadores e filtragens para refinar os resultados.

## 5 Diagrama da Padronização e Limpeza

- Entrada dos dados;
- Seleção dos campos;
- Padronização dos dados;
- Verificação dos nomes;
- Saída dos dados.

Os dados foram inicialmente recebidos por meio de um arquivo CSV disponível na pasta "Source". Após a padronização, a saída dos dados foi gerada em formato JSON e guardada na pasta "Data".

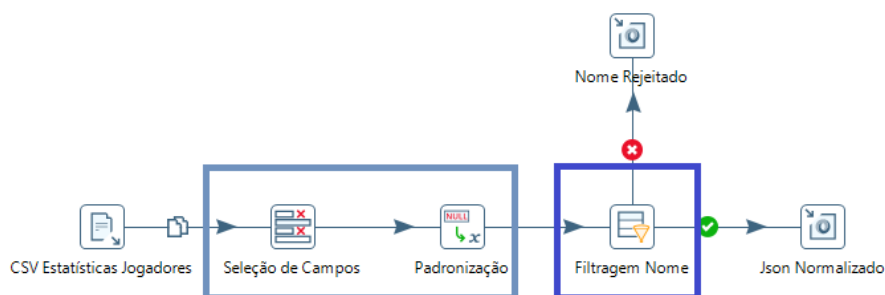


Figure 1: Fluxo da Primeira Transformação

### 5.1 Seleção e Padronização de Campos

No primeiro quadrado da imagem 1 *Transformação*, é realizada a padronização dos dados, dividida em dois **steps**:

- O **step Select Values** foi utilizado para seleccionar os campos relevantes e definir os tipos de dados adequados para cada campo.
- O **step If Field Value Is Null** foi configurado para lidar com valores nulos nos campos, substituindo-os por valores predefinidos.

Select values

Step name Seleção de Campos

Select & Alter Remove Meta-data

Fields to alter the meta-data for:

#	Fieldname	Rename to	Type	Length	Precision	Binary to Normal?	Format	Date Format Lenient?	Date Locale
1	Jersey_number		Integer			S		N	
2	Clean_sheets		Integer			S		N	
3	Goals_conceded		Integer			S		N	
4	Last_man_tackles		Integer			S		N	
5	Clearances_off_line		Integer			S		N	
6	Own_goals		Integer			S		N	
7	Tackle_success_percentage		BigNumber			S		N	
8	Recoveries		Integer			S		N	
9	Duels_won		Integer			S		N	
10	Duels_lost		Integer			S		N	
11	Successful_FiftyFiftys		Integer			S		N	
12	Aerial_battles_won		Integer			S		N	
13	Aerial_battles_lost		Integer			S		N	
14	Errors_leading_to_goal		Integer			S		N	
15	Cross_accuracy_percentage		BigNumber			S		N	

Figure 2: Seleção de Campos

Fields

#	Field	Replace by value	Conversion mask (Date)	Set empty string?
1	Jersey_number	-1		N
2	Clean_sheets	-1		N
3	Goals_conceded	-1		N
4	Last_man_tackles	-1		N
5	Clearances_off_line	-1		N
6	Own_goals	-1		N
7	Tackle_success_percentage	-1%		N
8	Recoveries	-1		N
9	Duels_won	-1		N
10	Duels_lost	-1		N
11	Successful_FiftyFiftys	-1		N
12	Aerial_battles_won	-1		N
13	Aerial_battles_lost	-1		N
14	Errors_leading_to_goal	-1		N
15	Cross_accuracy_percentage	-1%		N
16	Through_balls	-1		N

Figure 3: Tratamento de Valores Nulos

O uso dos **steps Select Values** e **If Field Value Is Null** garantiu a integridade dos dados, evitando problemas decorrentes de registros nulos durante a análise.

## 5.2 Verificação de Nomes

Para a verificação dos nomes dos jogadores no 2 quadrado da *1 Transformação*, foi utilizada uma expressão regular que detecta a presença de números no campo de nome, a fim de validar a conformidade desses dados como podemos ver na imagem *Verificação dos Nomes*.

Filter rows

Step name: Filtragem Nome

Send 'true' data to step: Json Normalizado

Send 'false' data to step: Nome Rejeitado

The condition:

Name REGEXP ^[^\d]+\$ (String)

Figure 4: Verificação dos Nomes



## 6 Diagrama de Filtragem e Cálculo de Performance

- Entrada dos dados;
- Filtragem Apperances;
- Cálculo Performance;
- Saída dos dados.

Os dados foram inicialmente recebidos por meio de um arquivo JSON disponível na pasta "Data". Após a filtragem e o cálculo, a saída dos dados foi gerada em formato XML e guardada na pasta "Data".

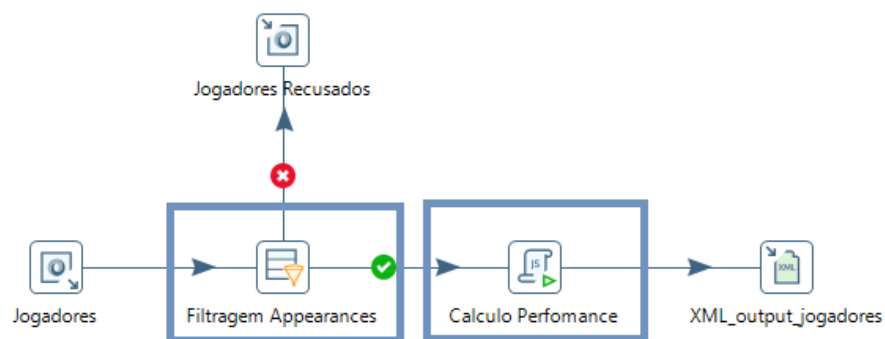


Figure 5: Filtragem e Cálculo de Performance

### 6.1 Filtragem

Neste **step**, foram filtrados os jogadores com base no número de partidas jogadas, garantindo que somente os jogadores com participação significativa nas competições fossem incluídos na análise.

Isto evita que jogadores com poucas aparições distorçam os resultados.

Filter rows

Step name

Filtragem Appearances

Send 'true' data to step:

Calculo Performance

Send 'false' data to step:

Jogadores Recusados

The condition:

Appearances

>

30

(Integer)

Figure 6: Filtragem por Número de Jogos

## 6.2 Cálculo de Performance

Neste **step**, foi calculada a pontuação de desempenho dos jogadores com base nas suas estatísticas, garantindo a inclusão apenas daqueles que tiveram participação significativa nas competições. Isto foi essencial para evitar distorções nos resultados causadas por jogadores com poucas aparições. A pontuação de desempenho reflete a contribuição relativa de cada jogador, utilizando critérios específicos para as suas respectivas posições no campo (os cálculos são fictícios, e alguns jogadores podem apresentar valores superiores a 100).

Step name

Calculo Performance

JavaScript

```

var performanceScore = 0;

// Função auxiliar para verificar se um valor é válido (não -1)
function isValid(stat) {
  return stat !== -1;
}

// Normaliza a posição para garantir que a comparação seja precisa
var normalizedPosition = Position.trim().toLowerCase();

// Cálculo para "Goalkeeper"
if (normalizedPosition === "goalkeeper") {
  performanceScore = (isValid(Saves) ? (Saves * 0.1) : 0)
    + (isValid(Clean_sheets) ? (Clean_sheets * 0.2) : 0)
    + (isValid(Wins) ? (Wins * 0.1) : 0)
    + (isValid(Penalties_saved) ? (Penalties_saved * 0.15) : 0)
    + (isValid(High_Claims) ? (High_Claims * 0.1) : 0)
    + (isValid(Sweeper_clearances) ? (Sweeper_clearances * 0.05) : 0)
    + (isValid(Goal_Kicks) ? (Goal_Kicks * 0.05) : 0)
    + (isValid(Punches) ? (Punches * 0.05) : 0)
    - (isValid(Yellow_cards) ? (Yellow_cards * 0.05) : 0)
    - (isValid(Red_cards) ? (Red_cards * 0.3) : 0)
    - (isValid(Losses) ? (Losses * 0.1) : 0)
    - (isValid(Errors_leading_to_goal) ? (Errors_leading_to_goal * 0.3) : 0)
    - (isValid(Goals_conceded) ? (Goals_conceded * 0.1) : 0);

// Cálculo para "Defender"
} else if (normalizedPosition === "defender") {
  performanceScore = (isValid(Tackles) ? (Tackles * 0.1) : 0)
    + (isValid(Clean_sheets) ? (Clean_sheets * 0.2) : 0)
    + (isValid(Interceptions) ? (Interceptions * 0.2) : 0)
    + (isValid(Aerial_battles_won) ? (Aerial_battles_won * 0.1) : 0)
    - (isValid(Yellow_cards) ? (Yellow_cards * 0.1) : 0)
    - (isValid(Red_cards) ? (Red_cards * 0.3) : 0)
    - (isValid(Errors_leading_to_goal) ? (Errors_leading_to_goal * 0.1) : 0)
    - (isValid(Goals_conceded) ? (Goals_conceded * 0.05) : 0)
    - (isValid(Accurate_long_balls) ? (Accurate_long_balls * 0.05) : 0)
    - (isValid(Duels_lost) ? (Duels_lost * 0.05) : 0);

// Cálculo para "Midfielder"
} else if (normalizedPosition === "midfielder") {
  performanceScore = (isValid(Assists) ? (Assists * 0.2) : 0)
    + (isValid(Passes) ? (Passes * 0.1) : 0);

```

Figure 7: Cálculo Performance Score

## 7 Diagrama de Conexão com Base de Dados

- Entrada dos dados;
- Replace in string;
- Saída dos dados.

Os dados foram inicialmente recebidos por meio de um arquivo XML disponível na pasta "DATA". Após a mudança nas strings, a saída dos dados foi gerada para uma base de dados **mysql**.

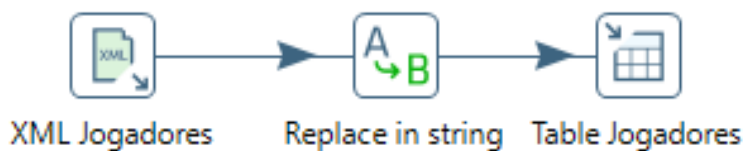


Figure 8: Conexão com Base de Dados

### 7.1 Substituição de Strings

Para preparar os dados para inserção na base de dados, foi necessário substituir as vírgulas nos campos com valores decimais por pontos.

#	In stream field	Out stream field	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive	Is Unicode
1	Cross_accuracy_percentage		N	,	.	N		N	N	N
2	Shooting_accuracy_percentage		N	,	.	N		N	N	N
3	Tackle_success_percentage		N	,	.	N		N	N	N
4	Passes_per_match		N	,	.	N		N	N	N
5	Goals_per_match		N	,	.	N		N	N	N
6	performanceScore		N	,	.	N		N	N	N

Figure 9: Substituição de Vírgulas por Pontos

## 8 Diagrama dos 5 Melhores Jogadores por Posição

- Entrada dos dados;
- Organização por Posição;
- Ordenação;
- Seleção de dados;
- Saída dos dados.

Os dados nesta última transformação vieram da base de dados **isi\_t01** . Após o devido tratamento dos dados, a saída dos dados foi gerada para o formato excel e guardada na pasta "Data".

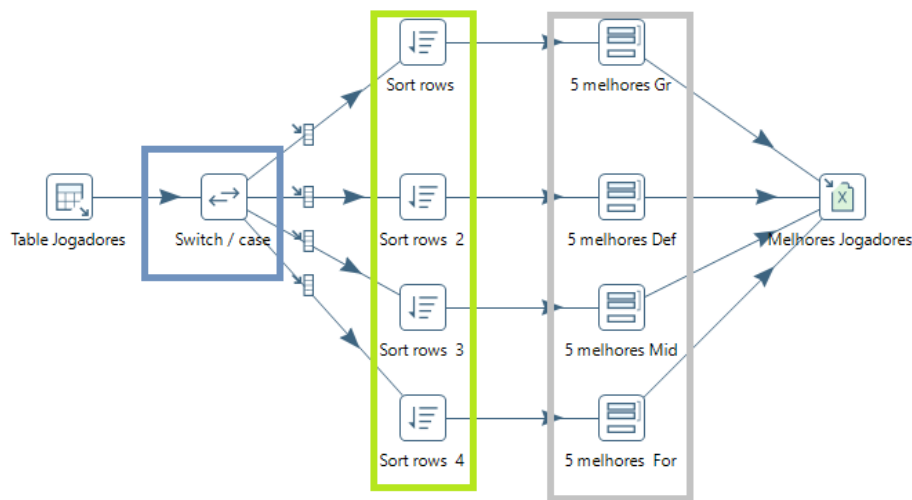


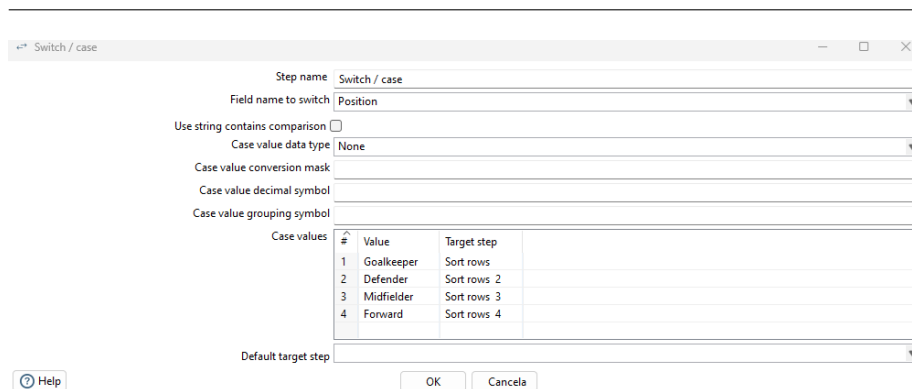
Figure 10: Seleção dos 5 Melhores Jogadores por Posição

### 8.1 Agrupamento por Posição

O **step Switch/Case** foi utilizado para organizar os dados dos jogadores de acordo com suas respectivas posições em campo.

Para cada posição, foi definida uma rota específica, assegurando que os dados fossem processados de forma independente e apropriada para cada caso.

Na imagem, é possível visualizar claramente o que foi descrito acima.

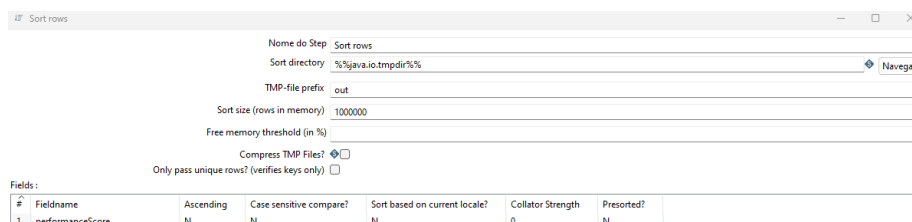


#	Value	Target step
1	Goalkeeper	Sort rows
2	Defender	Sort rows 2
3	Midfielder	Sort rows 3
4	Forward	Sort rows 4

Figure 11: Agrupamento por Posição

## 8.2 Ordenação dos Dados

O **step Sort Rows** foi utilizado para ordenar os registros dentro de cada grupo de dados gerado a partir do **Switch/Case**. A ordenação foi feita com base no campo `score_performance`, de forma a garantir que os jogadores fossem organizados de acordo com seu desempenho.



#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	performanceScore	N	N	N	0	N

Figure 12: Ordenação por Performance

## 8.3 Seleção de Dados

O **step Sample Rows** foi utilizado para selecionar as linhas necessárias para atingir o objetivo final de identificar os cinco melhores jogadores em cada posição. Por isso colocamos a **Lines range de 1 a 5**.

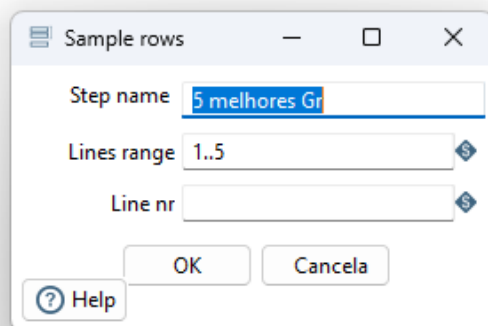


Figure 13: Seleção 5 melhores

## 9 Job

O job inicia com a verificação da existência do arquivo de entrada. Caso o arquivo seja encontrado, o processo continua com a criação do diretório de destino(**Data**), onde os resultados processados serão armazenados.

Após essa verificação inicial, o job entra na fase de padronização dos dados. Já explicada anteriormente.

Com a padronização concluída com sucesso, o job avança para a fase de filtragem e cálculo das métricas de desempenho dos jogadores. Esta fase é a fase essencial do projeto.

O próximo passo é a conexão com o banco de dados. Aqui, os dados já filtrados e processados são inseridos ou consultados na base de dados, permitindo o armazenamento ou a leitura das informações necessárias.

A próxima transformação consiste no agrupamento dos jogadores por posição e na classificação de acordo com sua performance num excel com os 5 melhores jogadores de cada posição.

Em seguida, é feito um ping ao host e, caso haja resposta, um email é enviado automaticamente ao utilizador. Após isso, a pasta **Data** criada anteriormente é excluída, finalizando o processo com sucesso..

Se ocorrer qualquer falha em uma das etapas, o job é interrompido no ponto do erro, e uma mensagem detalhada é enviada ao utilizador, especificando a origem do problema. Tanto em caso de sucesso quanto em caso de erro, o utilizador recebe uma notificação, assegurando a transparência durante toda a execução do job.

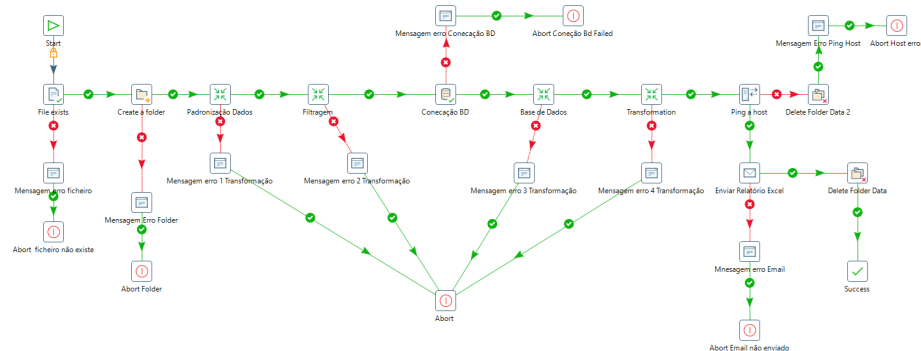


Figure 14: Job

## 10 Node-RED

O **Node-RED** oferece um conjunto robusto de nós dedicados à criação de dashboards interativos, facilitando a geração de gráficos e a visualização de dados em tempo real. Esta característica foi fundamental para a escolha desta ferramenta no projeto, devido à sua simplicidade e eficiência na apresentação visual dos resultados.

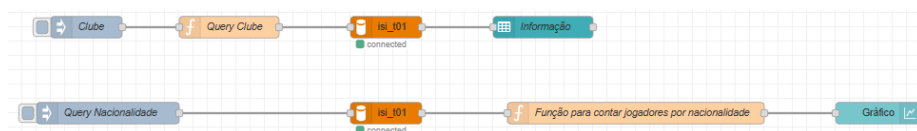


Figure 15: Node-Red

Utilizei o Node-RED para desenvolver dois processos principais:

1. **Gráfico de Nacionalidades:** Este gráfico exibe a distribuição das nacionalidades de todos os jogadores presentes na base de dados, permitindo uma visão clara da diversidade no campeonato.



Figure 16: Gráfico



2. **Tabela de Seleção por Clube:** Além do gráfico, criei uma tabela que lista os jogadores filtrados por um determinado clube. Esta tabela fornece o nome dos jogadores da equipa.

Jogadores	
Name	
Andreas Christensen	
Antonio Rüdiger	
Ben Chilwell	
Callum Hudson-Odoi	
César Azpilicueta	
Jorginho	
Kepa Arrizabalaga	
Kurt Zouma	
Marcos Alonso	
Mason Mount	
Mateo Kovacic	
N'Golo Kanté	
Olivier Giroud	
Ross Barkley	
Ruben Loftus-Cheek	
Tammy Abraham	
Willy Caballero	

Figure 17: Jogadores Chelsea

## 11 Vídeo com demonstração (QR Code)

Com o **QR Code** abaixo, tem uma demonstração em vídeo, a executar o **job** e a utilização do **Node-RED**.



Figure 18: QR-code

## 12 Conclusão

Este trabalho **ETL** proporcionou uma análise abrangente das estatísticas dos jogadores da Premier League, permitindo identificar os cinco melhores jogadores em cada posição, através da aplicação de diversas etapas de transformação e filtragem.

A normalização dos dados, a filtragem de jogadores com base na sua participação nas competições e o cálculo de métricas de desempenho foram essenciais para a análise. O uso dos steps adequados no **Kettle** facilitou a execução das transformações, otimizando o processo de **ETL**.

Além disso, a visualização dos dados finais demonstrou a eficácia da abordagem utilizada, permitindo a identificação clara dos jogadores que se destacaram em suas respectivas posições. Essa análise não apenas ajuda a compreender melhor o desempenho dos atletas, mas também serve como uma ferramenta valiosa para clubes e técnicos na tomada de decisões estratégicas.

Em suma, o trabalho realizado ilustra a importância de um processo de **ETL** bem estruturado na análise de dados esportivos, evidenciando como técnicas de manipulação de dados podem contribuir para insights significativos no contexto do futebol.

## 13 Bibliografia

<https://www.kaggle.com/datasets/rishikeshkanabar/premier-league-player-statistics-updated-daily/data>

<https://nodered.org/>