

Trabalho prático 1: ETL
Rúben Martins Ribeiro
Nº 25994 – Regime Pós-laboral

Ano letivo 2024/2025

Licenciatura em Engenharia de Sistemas Informáticos
Escola Superior de Tecnologia
Instituto Politécnico do Cávado e do Ave

Identificação do Aluno

[Rúben Martins Ribeiro]

Aluno número 25994, regime pós-laboral

Licenciatura em Engenharia de Sistemas Informáticos

RESUMO

Neste trabalho, foi realizado o tratamento para lidar com as inconsistências presentes numa tabela Excel formatada para CSV contendo informações sobre jogadores de futebol.

O processo de correção manual não resolveu adequadamente o problema da presença de erros e prejudicou diretamente a qualidade e a garantia da confiabilidade dos dados.

Para resolver esses erros, foi desenvolvida uma ferramenta KNIME que automatiza a verificação e a normalização das informações. Os fluxos de trabalho criados permitiram detetar erros, como a formatação incorreta e a duplicação dos dados, e disponibilizar as informações de forma correta e oportuna.

Além de garantir a qualidade e a confiabilidade dos dados, as melhorias aplicadas aumentaram a certeza na análise, reduzindo o tempo despendido.

A abordagem desenvolvida também teve resultados que podem ser utilizados para futuras análises de maneiras muito simples, incluindo as correções aplicadas nos dados dos jogadores.

ABSTRACT

In this work, I dealt with the inconsistencies present in a CSV-formatted Excel table containing information on soccer players.

The manual correction process did not adequately solve the problem of the presence of errors and directly affected the quality and reliability of the data.

To resolve these errors, a KNIME tool was developed to automate the verification and normalization of information. The workflows created made it possible to detect errors, such as incorrect formatting and duplicate data, and to make the information available correctly and in a timely manner.

As well as guaranteeing the quality and reliability of the data, the improvements applied have increased certainty in the analysis, reducing the time spent.

The approach developed also yielded results that can be used for future analysis in very simple ways, including the corrections applied to the players' data.

ÍNDICE

1. INTRODUÇÃO	10
1.1. Objetivos	10
1.2. Contexto	10
1.3. Estrutura do documento	11
2. IMPLEMENTAÇÃO	12
2.1. Demonstração do fluxo geral	12
2.2. Importação dos dados dos jogadores	13
2.3. Processo de correção de dados	14
2.4. Processo de exportação de dados	21
2.4.1. Exportação para JSON	21
2.4.2. Exportação para CSV	22
2.4.3. Exportação para XLSX	23
3. CONCLUSÃO	27
4. BIBLIOGRAFIA	28

ÍNDICE DE FIGURAS

FIGURA 1: VISÃO DO PROJETO NO KNIME	12
FIGURA 2: IMPORTAÇÃO DO ARQUIVO CSV DE JOGADORES	13
FIGURA 3: PROCEDIMENTO CORREÇÃO DE IDS	14
FIGURA 4: RESULTADO FINAL DOS IDS	15
FIGURA 5: UTILIZAÇÃO DE EXPRESSÃO REGULAR	15
FIGURA 6: CÁLCULO DA IDADE ATUAL	19
FIGURA 7: REPETIÇÃO DO PROCESSO INICIAL	19
FIGURA 8: SEQUENCIA FINAL	21
FIGURA 9: FILTRAGEM DO JOGADOR	22
FIGURA 10: CONTAGEM DE CAMISOLAS	23
FIGURA 11: EXPORTAÇÃO DOS FICHEIROS	23
FIGURA 12: EXPORTAÇÃO PARA FICHEIRO JSON	24
FIGURA 13: EXPORTAÇÃO PAR FICHEIRO XLSX	24
FIGURA 14: EXPORTAÇÃO PAR FICHEIRO CSV	25

ÍNDICE DE TABELAS

TABELA 1:CORREÇÃO DE IDS	15
TABELA 2:CORREÇÃO DE SÍMBOLOS DESCONHECIDOS	16
TABELA 3:CORREÇÃO DA COLUNA AGE	16
TABELA 4: ABREVIATURA DE CLUBES	20
TABELA 5: CORREÇÃO NA COLUNA NATIONALITY	22

Glossário

Excel – Software desenvolvido pela Microsoft, amplamente utilizado para manipulação e análise de dados.

JavaScript – Linguagem de programação, amplamente utilizada para criar interatividade em páginas web.

String – Tipo de dado utilizado em programação para representar uma sequência de caracteres.

KNIME – Plataforma de análise de dados de código aberto que permite a criação de fluxos de trabalho visuais para extração, transformação e análise de dados.

Data Cleaning: Procedimento de detecção e correção de dados incorretos, incompletos ou duplicados para assegurar a qualidade e consistência das informações no dataset.

Regular Expression (Regex): Sequência de caracteres que define um padrão de pesquisa em texto, utilizada para localizar, substituir ou formatar informações dentro dos dados, como na remoção de caracteres indesejados.

Data Normalization: Processo de padronização dos dados para garantir consistência e eliminar redundâncias, fundamental para a integridade das análises.

Data Export: Etapa final do fluxo de trabalho de ETL, onde os dados tratados são armazenados em formatos específicos, como CSV, JSON ou XLSX, de acordo com as necessidades do sistema de destino.

Workflow: Conjunto de operações ou etapas estruturadas em um fluxo sequencial. No KNIME, refere-se à organização de nós que executam tarefas específicas no processo de tratamento de dados.

Siglas e Acrónimos

CSV - Comma-Separated Values (Valores Separados por Vírgula), um formato de ficheiro simples e amplamente usado para armazenamento de dados tabulares.

ETL - Extract, Transform, Load (Extrair, Transformar, Carregar), um processo utilizado para a integração e limpeza de dados.

JSON - JavaScript Object Notation, um formato leve de intercâmbio de dados em texto.

KNIME - Konstanz Information Miner, uma plataforma de análise de dados que permite criar workflows para automação de ETL e processamento de dados.

XLSX - Formato padrão de arquivos do Microsoft Excel que permite salvar e organizar dados em planilhas.

1. Introdução

A área de Sistemas Informáticos desempenha um papel importante na gestão e análise de dados em diversas aplicações, sendo essencial para a obtenção de decisões informadas. Com o aumento da quantidade de dados disponíveis, a necessidade de ferramentas que permitem a correção e normalização de informações tornou-se cada vez mais recorrente.

Neste contexto, os formatos de ficheiros como CSV são bastante utilizados para a troca de dados, especialmente em ambientes empresariais ou académicos. No entanto, a utilização desses dados pode frequentemente resultar em inconsistências e erros, afetando a qualidade das análises realizadas.

Este trabalho baseia-se na correção de erros encontrados numa tabela Excel no formato CSV que continha informações sobre jogadores de futebol. A má identificação e a inconsistência dos dados representam um desafio significativo, pois comprometem a fiabilidade das informações e consequentemente, das decisões baseadas nelas.

Para abordar esse problema, utilizei a ferramenta KNIME, que possibilita a automatização de processos de ETL de forma eficiente. O objetivo foi identificar e corrigir os erros presentes na base de dados, garantindo a integridade e a qualidade das informações dos jogadores.

1.1. Objetivos

- Identificação de erros: Reconhecer e classificar os erros na tabela de dados sobre os jogadores de futebol.
- Correção de dados: utilizar as técnicas de normalização e limpeza de dados para garantir que as informações sejam corretas e adequadas.
- Automatização de Processo: Utilizar a ferramenta KNIME para automatizar o processo de verificação, correção e economizar tempo e esforço neles investidos.
- Validação de Resultado: Garantir que os dados corrigidos possam ser usados em análises e relatórios futuros.
- Documentação do Processo: Produzir um relatório bem estruturado apresentando as etapas e resultados para esclarecer a importância da qualidade dos dados na tomada de decisão informada.

1.2. Contexto

O projeto foi desenvolvido com o objetivo de corrigir inconsistências numa tabela CSV que contém informações sobre jogadores de futebol.

Dada a importância da qualidade dos dados para a análise de desempenho, este trabalho insere-se num contexto onde erros e duplicações podem afetar a confiabilidade das análises realizadas.

Através da ferramenta KNIME, foram aplicados processos de ETL para automatizar a verificação, normalização e limpeza dos dados, assegurando a integridade e fiabilidade para futuras análises.

1.3. Estrutura do documento

Introdução

- **Objetivos:** Delimitação do propósito do projeto, que inclui identificar e corrigir inconsistências numa base de dados de jogadores de futebol.
- **Contexto:** Explicação do uso de dados no formato CSV e da importância de garantir a qualidade dos dados no contexto de análise desportiva.
- **Estrutura do Documento:** Apresentação da organização do relatório, incluindo as etapas principais de implementação e análise dos resultados.

Análise do Problema

- **Inconsistências no Dataset:** Descrição das principais falhas encontradas na base de dados, tais como duplicidade de registos e formatação incorreta.
- **Impacto dos Erros na Qualidade dos Dados:** Discussão sobre como dados de baixa qualidade podem afetar decisões analíticas e estratégicas no contexto desportivo.
- **Revisão de Ferramentas de ETL:** Introdução breve às principais ferramentas e metodologias aplicáveis ao tratamento de dados desportivos.

Análise e Modelação do Sistema

- **Diagrama de Fluxo do Workflow em KNIME:** Representação visual do fluxo de operações utilizado para extrair, transformar e carregar os dados.
- **Componentes do Processo ETL:** Descrição dos nós e operações específicos em KNIME necessários para atingir os objetivos do sistema.

Implementação

- **Importação dos Dados:** Detalhes sobre o processo de importação da tabela CSV no KNIME, incluindo a configuração inicial do delimitador de dados.
- **Processo de Correção de Dados:** Etapas de verificação e correção de IDs e manipulação de strings para garantir a padronização das informações.
- **Normalização e Exportação dos Dados:** Procedimento de exportação dos dados em formatos JSON, XLSX e CSV, com uma explicação das finalidades de cada formato.

Análise de Resultados e Testes

- **Comparação de Dados Pré e Pós-Processamento:** Resultados obtidos em termos de correção de erros e eficiência de análise após o processo de ETL.
- **Usabilidade do Sistema Desenvolvido:** Análise do impacto da automatização no tratamento de dados desportivos e possíveis melhorias.

Conclusão

- **Resumo dos Principais Resultados:** Recapitulação das melhorias obtidas em termos de eficiência, precisão e qualidade dos dados alcançadas com o workflow implementado.
- **Impacto da Qualidade dos Dados na Análise Desportiva:** Reflexão sobre a importância de dados corretos na tomada de decisões analíticas.
- **Trabalhos Futuros e Melhorias:** Sugestões para futuras otimizações e possibilidades de expansão do sistema.

Bibliografia

- **Lista das Referências Utilizadas:** Relação das fontes e referências utilizadas para embasar o projeto, incluindo documentação técnica, artigos e livros sobre processos de ETL e o uso da plataforma KNIME.

2. Implementação

Para este trabalho, utilizei a plataforma KNIME, uma ferramenta de análise de dados que permitiu desenvolver uma solução para automatizar o processo completo que abrange desde a recção até a preparação final dos dados para uso.

Integrando a linguagem JavaScript adaptada à aplicação e com a utilização de expressões regulares, foi possível implementar um processo completo de normalização, filtragem, correção de erros e exportação, melhorando cada etapa para garantir a qualidade e consistência dos dados.

2.1. Demonstração do fluxo geral

Esta figura mostra o funcionamento geral do projeto no KNIME. Nela, destacam-se as etapas do processo ETL, incluindo a leitura, limpeza, transformação e exportação dos dados. Explica o propósito do projeto e como o funcionamento foi estruturado para resolver as inconsistências do arquivo CSV original.



Figura 1: Visão do projeto no Knime

2.2. Importação dos dados dos jogadores

Para iniciar o processo de tratamento dos dados no KNIME, realizei a importação da tabela CSV localizada numa pasta específica. Utilizei o nó de leitura de CSV para aceder ao ficheiro diretamente na pasta, garantindo a localização correta dos dados iniciais. Durante a configuração, defini o delimitador do ficheiro como ponto e vírgula (;), o que permitiu ao KNIME interpretar corretamente as colunas do ficheiro original e segmentar os dados conforme a estrutura do CSV.

Após esta configuração, o KNIME converteu automaticamente as informações do CSV numa tabela de dados, apresentando-as em colunas separadas para facilitar as próximas etapas de tratamento e normalização.

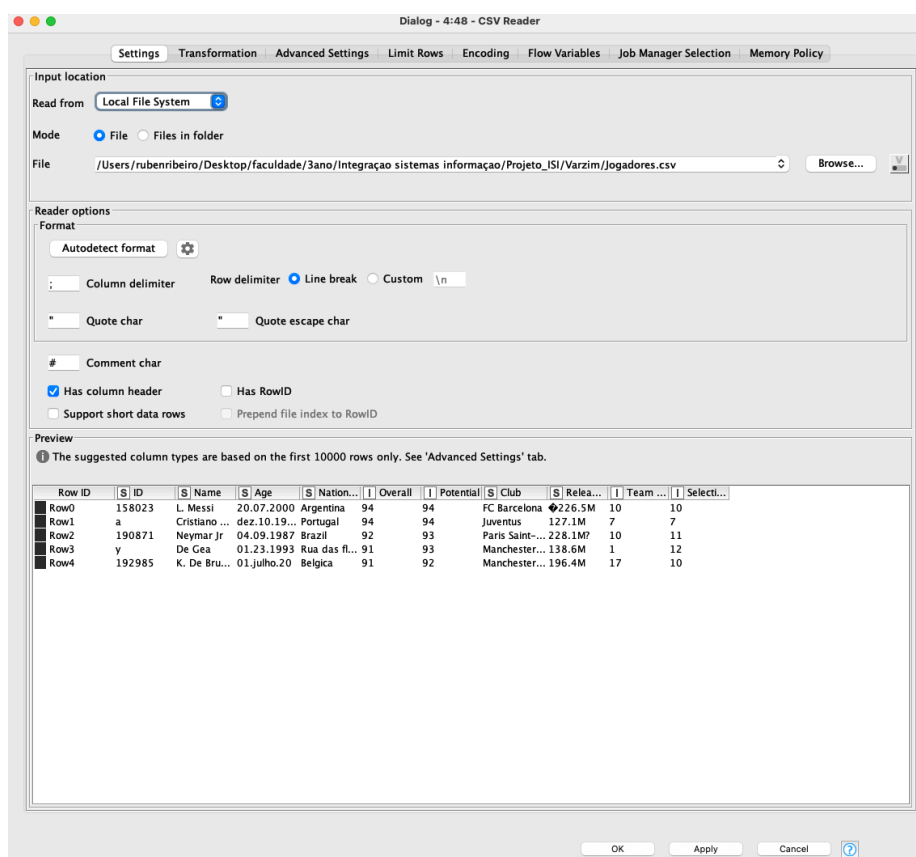


Figura 2: Importação do arquivo CSV de jogadores

2.3. Processo de correção de dados

Na Figura 3, o processo de correção dos IDs é detalhado. Utilizei o nó *Column Expressions* para inserir o código responsável por identificar e corrigir os erros nos IDs presentes na coluna, criando uma nova coluna ID_new com os IDs corretos .

Em seguida, apliquei o nó *Column Filter* para isolar a coluna ID, onde estavam localizados os erros, garantindo que a coluna fosse extraída com os mesmos erros.

Após a extração e correção, organizei os dados utilizando o nó *Column Resorter* para reposicionar a coluna ID corrigida no local adequado. Em seguida, com o nó *Column Renamer*, ajustei o nome da coluna de ID_new para ID, restabelecendo o nome original para assegurar a continuidade do fluxo com a nova coluna corrigida. Na parte inferior(figura 4), apresenta-se o resultado final com os IDs ajustados corretamente.



Figura 3: Procedimento correção de Ids

Neste código, utilizei uma lógica condicional para corrigir os valores na coluna ID, assegurando que todos os dados ficassem no formato numérico adequado. A primeira condição verifica se que o primeiro valor na coluna ID é "a". Quando isso acontece, substituo este valor pelo ID "20801". Em seguida, verifico se o valor é "y", e neste caso, faço a substituição pelo ID "193080". Estas condições iniciais resolvem casos específicos em que o ID não está numérico e necessita de correção manual.

Para os restantes valores na coluna ID (ou seja, quando o valor não é "a" nem "y"), aplico uma expressão regular com a função `regexReplace`. Esta função remove todos os caracteres que não são numéricos, utilizando a expressão `^[^0-9]` para identificar e eliminar caracteres indesejados, tal como no caso do ID 1, que inclui o símbolo `&`, e do ID 5, que contém a letra "e" e o símbolo "?".

Em seguida, utilizei o `parseInt` para converter a sequência resultante numa variável numérica (int). Desta forma, garanto que todos os IDs estejam no formato correto como números inteiros, independentemente dos caracteres incorretos que possam ter originalmente.

```

if (column("ID") == "a") {
id = "20801";
} else if (column("ID") == "y") {
id = "193080";
} else {

id = parseInt(regexReplace(string(column("ID")), "[^0-9]", ""));
}

```

Tabela 1: Correção de ids

#	RowID	ID <small>Number (integer)</small>	Name <small>String</small>	Age <small>String</small>	Nationality <small>String</small>	Overall <small>Number (integer)</small>	Potential <small>Number (integer)</small>	Club <small>String</small>	Release Clause <small>String</small>	Team Number <small>Number (integer)</small>	Selection Num... <small>Number (integer)</small>
1	Row0	158023	L. Messi	20.07.2000	Argentina	94	94	FC Barcelona	226.5M	10	10
2	Row1	20801	Cristiano Ronaldo	dez.10.1997	Portugal	94	94	Juventus	127.1M	7	7
3	Row2	190871	Neymar Jr	04.09.1987	Brazil	92	93	Paris Saint-Germain	228.1M?	10	11
4	Row3	193080	De Gea	01.23.1993	Rua das flores	91	93	Manchester United	138.6M	1	12
5	Row4	192985	K. De Bruyne	01.julho.20	Belgica	91	92	Manchester City	196.4M	17	10

Figura 4: Resultado final dos Ids

Na Figura 5, são apresentados dois nós String Manipulation utilizados para corrigir dados incorretos em certas colunas.

Primeiro, utilizei uma operação de manipulação de strings para substituir valores indesejados ou formatar corretamente os dados onde pode incluir a remoção de espaços em branco, substituição de caracteres especiais ou a padronização de formatos de texto.

Em seguida, apliquei uma segunda operação de manipulação de strings que complementou a primeira, abordando outras inconsistências específicas que poderiam interferir nas análises futuras. Por exemplo, se havia texto não padronizado ou formatação inconsistente nas colunas, essas operações garantiram que todos os dados estivessem uniformes e prontos para uso.

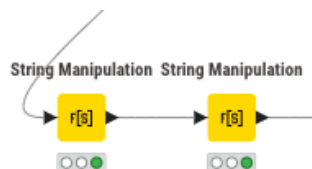


Figura 5: Utilização de expressão regular

Na tabela 2, estou a utilizar a função `regexReplace` para limpar os dados incorretos da coluna "Release Clause". O objetivo é remover todos os caracteres indesejados, garantindo que a informação na coluna contenha apenas letras, números e pontos. A expressão regular `^[^a-zA-Z0-9.]` é o elemento central onde o símbolo `^` no início da expressão indica uma negação, ou seja, estou a dizer que queremos eliminar tudo que não está dentro dos parênteses. As partes `a-z` e `A-Z` referem-se a todas as letras minúsculas e maiúsculas do alfabeto, enquanto `0-9` abrange todos os dígitos numéricos. O ponto `.` é incluído na lista de caracteres permitidos no contexto da cláusula de rescisão, pois é comum ver valores decimais. Assim, ao aplicar a função `regexReplace($Release Clause$, "[^a-zA-Z0-9.]", "")`, estou a substituir qualquer caractere que não seja uma letra, um número ou um ponto por uma string vazia. Isto resulta numa versão limpa da coluna "Release Clause", onde todos os símbolos especiais e espaços são removidos, facilitando a análise posterior dos dados.

```
regexReplace($Release Clause$, "[^a-zA-Z0-9.]", "")
```

Tabela 2: correção de símbolos desconhecidos

Neste código, estou a utilizar uma série de chamadas à função `regexReplace` para normalizar e corrigir os dados na coluna "Age". O objetivo é substituir abreviações de meses e ajustar o formato das datas de nascimento para um padrão normal.

O primeiro passo consistiu em substituir os nomes dos meses por números correspondentes. Para isso, aplico várias chamadas à função `regexReplace`, começando por "jan", que é substituído por "01", e assim por diante, até "dez", que se torna "12". Uma vez que o mês de julho estava escrito por inteiro tive de ajustar o código manualmente para corrigir esse mesmo erro, e qualquer mês que estivesse escrito por inteiro terá de se corrigir o código manualmente. Desta forma, todas as menções dos meses estão agora representadas de forma numérica, facilitando a análise posterior.

Após esta transformação, o código inclui uma substituição adicional que altera o ano de "20" para "2004", utilizando a expressão `.20$` para assegurar que apenas o final da string seja alterado.

Finalmente, a expressão `^(\\d{2})\\. (23)\\. (\\d{4})$` é utilizada para identificar um formato específico de data. Se a data estiver no formato "xx.23.yyyy", ela será reformulada para "xx.02.yyyy". Essa resolução foi feita manualmente, pois o processo é específico para esses casos. Isso assegura que as datas estejam consistentes e corretas no formato desejado.

Ao aplicar essa série de transformações, garantimos que os dados na coluna "Age" estejam devidamente normalizados, o que é essencial para análises precisas e para evitar inconsistências nos dados.

```
regexReplace(regexReplace(regexReplace(regexReplace(regexReplace(regexReplace(regexReplace(
regexReplace(regexReplace(regexReplace(regexReplace(regexReplace(regexReplace($Age$,
"jan", "01"), "fev", "02"), "mar", "03"), "abr", "04"), "mai", "05"), "jun", "06"), "julho", "07"), "ago", "08"),
"set", "09"), "out", "10"), "nov", "11"), "dez", "12"), ".20$", ".2004"), "^(\\d{2})\\. (23)\\. (\\d{4})$",
"$1.02.$3")
```

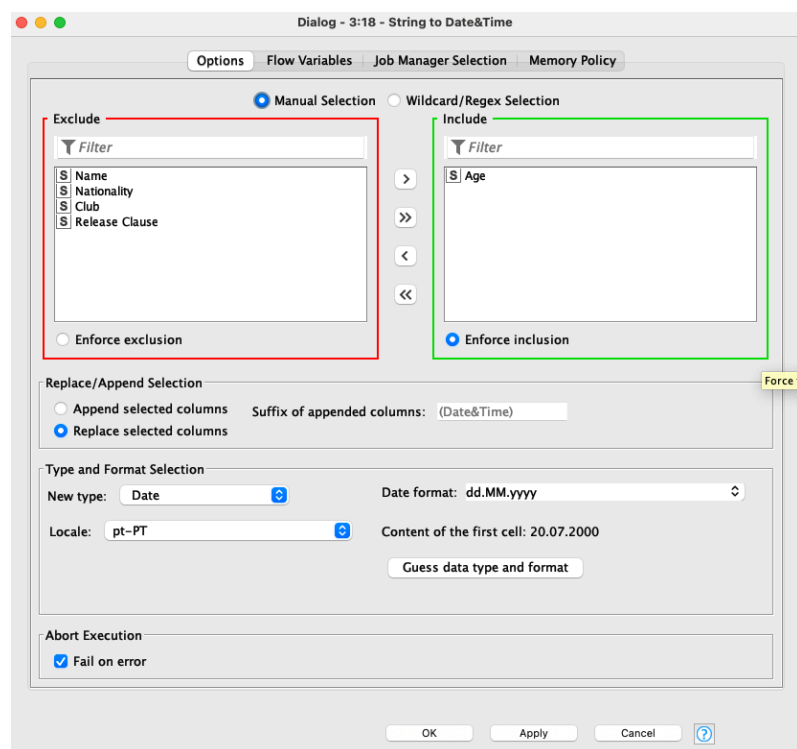
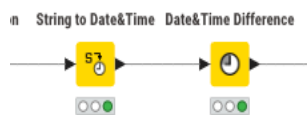
Tabela 3: Correção da coluna Age

Na figura 6, está representada a conversão e cálculo da idade exata de cada jogador, utilizando a coluna Age que se encontrava num formato específico.

Para realizar esta tarefa, comecei por isolar a coluna que continha as datas de nascimento (Age). O primeiro passo foi garantir que as datas estivessem no formato adequado, o que facilitou o processamento.

Utilizei o nó string Date&Time para aplicar os filtros e transformar os dados da coluna de data. Quando a data apresentava um formato específico (“xx.xx.xxx”), procedi às devidas conversões para garantir que todas as entradas fossem consistentes.

Após isolar a coluna Age, calculei a idade de cada jogador em anos, utilizando o nó Date&Time Difference. Este cálculo foi realizado com base na data de nascimento, subtraindo essa data da data atual. Este processo assegurou que as idades fossem calculadas com precisão, considerando anos, meses e dias, permitindo um resultado mais exato do que uma simples subtração de anos.



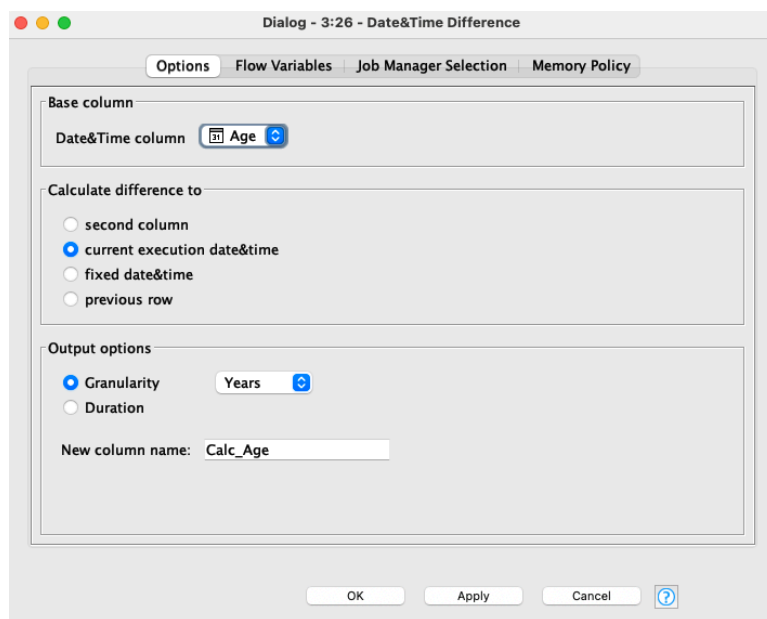


Figura 6: Cálculo da idade atual

No processo descrito mostrado em baixo, repeti a etapa inicial, onde extraí a tabela Age e reposicionei a tabela Calc_Age, renomeando-a posteriormente para Age.



Figura 7: Repetição do processo inicial

Na tabela 4 utilizei o nó *Column Expressions* para inserir o código responsável que cria uma versão abreviada dos nomes dos clubes de futebol a partir da coluna "Club". A variável *abbreviatedClub* é definida com base no valor presente na coluna.

O processo inicia-se por verificar se o nome do clube é "FC Barcelona"; caso seja, a variável é atribuída como "FCB". Em seguida, o código prossegue a verificar outros clubes, como "Juventus" que tem 2 vezes o mesmo nome, logo foi corrigida para que não haja duplicidade de dados, "Paris Saint-Germain", "Manchester United" e "Manchester City", atribuindo-lhes as respectivas abreviações.

Se o nome do clube não corresponder a nenhum dos clubes listados, a variável *abbreviatedClub* assume o nome completo do clube. Deste modo, esta lógica garante que os clubes reconhecidos tenham uma forma abreviada, enquanto aqueles que não estão na lista mantêm o seu nome original.

Este procedimento é bastante útil para tornar a análise dos dados mais eficiente e mais simples, uma vez que nomes mais curtos podem facilitar a leitura e interpretação das informações.



```
if (column("Club") == "FC Barcelona") {  
    abbreviatedClub = "FCB";  
} else if (column("Club") == "JuventusJuventos") {  
    abbreviatedClub = "JUV";  
} else if (column("Club") == "Paris Saint-Germain") {  
    abbreviatedClub = "PSG";  
} else if (column("Club") == "Manchester United") {  
    abbreviatedClub = "MU";  
} else if (column("Club") == "Manchester City") {  
    abbreviatedClub = "MC";  
} else {  
    abbreviatedClub = column("Club");  
}
```

Tabela 4: Abreviatura de clubes

2.4. Processo de exportação de dados

Na figura 9, podemos observar um exemplo em que, a partir do nó "Column Expression", são geradas três saídas distintas que terminam em diferentes tipos de ficheiros de escrita: **JSON Writer**, **CSV Writer** e **Excel Writer**.

Cada um desses nós de escrita é responsável por salvar os dados de maneira específica, de acordo com o formato de ficheiro escolhido, e cada um deles contém dados diferentes, resultado de processos anteriores realizados.

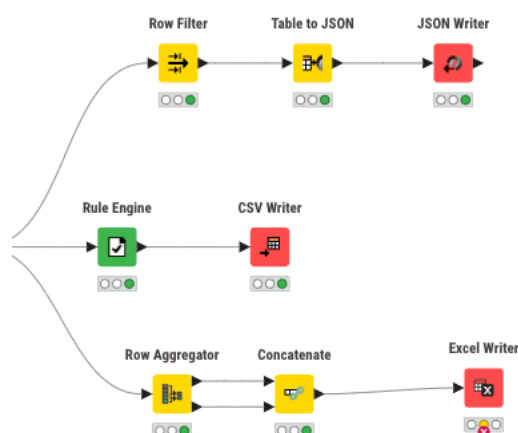


Figura 8: Sequencia final

2.4.1. Exportação para JSON

Na figura 10, apresento um exemplo do primeiro processo que resulta na exportação de um ficheiro JSON. Começo com o nó **Row Filter**, que tem como função isolar um jogador específico. Neste passo, estabeleci um critério de filtragem que exclui todos os outros jogadores, permitindo que apenas os dados do jogador escolhido sejam mantidos.

Após a filtragem, os dados finais são encaminhados para o nó **Table to JSON**. Este nó converte a tabela filtrada em formato JSON, o que é essencial para garantir que os dados sejam estruturados de uma forma que possa ser facilmente utilizada em outras aplicações ou sistemas.

Finalmente, a saída do nó **Table to JSON** é conectada ao **JSON Writer**. Este componente é responsável pela exportação final dos dados para um ficheiro no formato JSON. Esse procedimento permite que os dados do jogador específico que filtrei inicialmente sejam armazenados de maneira organizada, facilitando assim futuras análises ou integrações com outras plataformas.

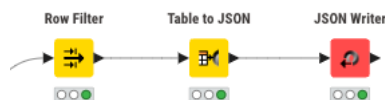


Figura 9: Filtragem do jogador

2.4.2. Exportação para CSV

No segundo procedimento, utilizei o nó Rule Engine para corrigir um erro na coluna de nacionalidade, onde uma entrada estava incorretamente registada como "Rua das Flores".

Após implementar essa correção, os dados ajustados foram encaminhados para o CSV Writer. Este nó é responsável por exportar os dados finais da tabela principal em formato CSV. Assim, todos os dados relevantes e corrigidos ao longo do processo foram armazenados de forma organizada neste ficheiro CSV.



A primeira parte da expressão, `Nationality$ = "Rua das Flores" => "Spain"`, é uma condição que verifica se o valor da coluna `Nationality` é igual a "Rua das Flores". Se essa condição for verdadeira, o código substituirá esse valor por "Spain". Este processo, sendo único e específico, transforma uma entrada incorreta (uma rua) na sua designação correta de país.

A segunda parte, `TRUE => $Nationality$`, serve como uma instrução que cobre todos os outros casos não especificados anteriormente. O uso de `TRUE` aqui indica que, se a condição anterior não se aplicar (ou seja, se o valor da nacionalidade não for "Rua das Flores"), o sistema deve manter o valor original da coluna `Nationality`. Essa abordagem garante que, para todas as nacionalidades que não precisam de correção, os dados permanecem inalterados.

<code>\$Nationality\$ = "Rua das flores" => "Spain"</code> <code>TRUE => \$Nationality\$</code>
--

Tabela 5: Correção na coluna `Nationality`

2.4.3. Exportação para XLSX

Por último, utilizei o nó Row Aggregator para calcular a quantidade de jogadores que tinham o mesmo número na camisola de cada equipa. Este processo consiste em agrupar os dados com base no número da camisola, permitindo contar quantas vezes cada número aparece na tabela. Por exemplo, se um jogador tiver o número 10 e outro também tiver o número 10, a contagem para o número 10 será de 2. De igual forma, se um jogador tiver o número 7 e outro tiver o número 8, a contagem resultante será de uma camisola de 7 e uma camisola de 8, sem somar as camisolas diferentes.

Após realizar essa agregação, conectei as tabelas resultantes para compilar todas as contagens de forma organizada, o que proporciona uma visão clara da distribuição das camisolas. Por fim, exportei essas informações para um ficheiro utilizando o nó Excel Writer, garantindo que a contagem final das camisolas fosse salva num formato acessível e fácil de manipular.

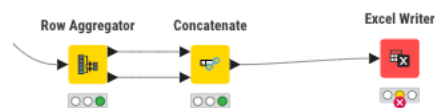


Figura 10:Contagem de camisolas

Por fim nesta figura estão a representados os ficheiros extraídos/encaminhados numa pasta específica onde guardei os mesmos



	Dados_Jogador_0.json	Ontem, 23:33	2E
	Dados_shirts.xlsx	Ontem, 20:14	
	Dados.csv	Ontem, 23:33	4C

Figura 11:Exportação dos ficheiros

Esta imagem mostra o processo de exportação dos dados(do jogador filtrado) tratados para o formato JSON, usando o nó *JSON Writer* no KNIME. Este formato é adequado para integração de dados em aplicações web e sistemas de

análise que suportam JSON.



```

{
  "Row2" : {
    "ID" : 190871,
    "Name" : "Neymar Jr",
    "Age" : 37,
    "Nationality" : "Brazil",
    "Overall" : 92,
    "Potential" : 93,
    "Club" : "PSG",
    "Release Clause" : "228.1M",
    "Team Number" : 10,
    "Selection Number" : 11
  }
}

```

Figura 12: Exportação para ficheiro JSON

A figura 14 representa a exportação dos dados(contagem de camisolas) para um arquivo Excel (XLSX) usando o nó *Excel Writer*. Este formato facilita o armazenamento e a análise dos dados em plataformas que utilizam planilhas.

Team Num	OCCURRENCE_COUNT
1	1
7	1
10	2
17	1
	5

Figura 13: Exportação par ficheiro XLSX

Nesta figura, observa-se o uso do nó *CSV Writer* para salvar os dados(Dados finais e corretos do projeto) em formato CSV, ideal para transferência e importação de dados entre diferentes sistemas e softwares de análise.

Dados

ID	Name	Age	Nationality	Overall	Potential	Club	Release Clause	Team Number	Selection Number
158023	L. Messi	24	Argentina	94	94	FCB	226.5M	10	10
20801	Cristiano Ronaldo	27	Portugal	94	94	JUV	127.1M	7	7
190871	Neymar Jr	37	Brazil	92	93	PSG	228.1M	10	11
193080	De Gea	31	Spain	91	93	MU	138.6M	1	12
192985	K. De Bruyne	20	Belgica	91	92	MC	196.4M	17	10

Figura 14: Exportação par ficheiro CSV

3. Conclusão

A execução deste trabalho prático evidenciou a importância da implementação de um processo de *Extract, Transform, Load* (ETL) para assegurar a consistência e a confiabilidade dos dados manipulados. Utilizando a plataforma KNIME, foi possível automatizar tarefas essenciais, como a detecção e correção de erros, a normalização de valores inconsistentes e a limpeza de dados, garantindo assim que as informações sobre jogadores de futebol fossem precisas e uniformes.

A automação das etapas de verificação e correção de dados não apenas eliminou a necessidade de validações manuais, mas também reduziu significativamente o tempo necessário para o tratamento das informações. Este ganho de eficiência minimizou o risco de erros humanos e assegurou que os dados fossem processados de maneira consistente e confiável. As operações incluíram a padronização de IDs, o cálculo de idades na coluna "Age", a correção de formatações incorretas como na coluna "Age", a abreviação do nome dos clubes para uma melhor leitura e compreensão e a aplicação de filtros e expressões regulares para remover caracteres indesejados em colunas específicas.

Além disso, o processo de exportação dos dados corrigidos em formatos variados, como JSON, XLSX e CSV, ampliou a usabilidade dos resultados para diferentes sistemas e aplicações de análise, contribuindo para a flexibilidade e adaptabilidade dos dados para futuras análises.

Este trabalho sublinha como uma abordagem estruturada de ETL, implementada no KNIME, pode transformar dados brutos e potencialmente problemáticos em informações valiosas e de fácil acesso, possibilitando a realização de análises mais robustas e fundamentadas.

4. Bibliografia

□ **Berthold, M., et al.** *KNIME - The Konstanz Information Miner: Version 2.0 and Beyond*. ACM SIGKDD Explorations, 11(1), 2009.

Descrição da plataforma KNIME e as suas funcionalidades de modelagem de dados e processos ETL.

KNIME Documentation. *KNIME Workflows and Nodes*.

Documentação oficial da KNIME com exemplos detalhados de workflows.

Zikopoulos, P., et al. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill, 2012.

Discussão sobre práticas de processamento de grandes volumes de dados, com foco em ETL e integração de dados.

KNIME Blog. "The New Date & Time Integration."

Artigo que aborda configurações de Date & Time no KNIME.

Rahm, E., & Do, H. H. *Data Cleaning: Problems and Current Approaches*. IEEE Data Engineering Bulletin, 23(4), 2000.

Estudo sobre técnicas de limpeza de dados, aplicáveis em ferramentas como o KNIME.