Relatório de Trabalho Prático I

Integração de Sistemas Informação

Autor:

- Pedro Rei (26013)

Licenciatura em Engenharia de Sistemas Informáticos
(3º ano)

Braga | Outubro, 2024

Índice

1. Introdução

No âmbito da unidade curricular de Integração de Sistemas de Informação, o presente trabalho tem como objetivo explorar e implementar processos de ETL (Extract, Transform, Load), essenciais para a integração eficaz de sistemas no que diz respeito à gestão de dados.

A crescente necessidade das organizações em consolidar e analisar grandes volumes de informações provenientes de diversas fontes torna imperativa a adoção de soluções eficientes que garantam a integridade e a consistência dos dados, além de facilitarem a tomada de decisões. Nesse contexto, os processos de ETL desempenham um papel crucial, uma vez que permitem a extração de dados de múltiplas origens, a sua transformação em formatos compatíveis e a carga subsequente em sistemas de destino.

Este relatório abordará as diferentes fases envolvidas na criação de um processo de ETL utilizando a ferramenta Pentaho Kettle na resolução de um problema real, destacando os desafios enfrentados ao longo do percurso e os resultados obtidos.

2. Descrição do Problema

Na cidade de Famalicão, com uma população de 10 mil pessoas, os dados principais dos habitantes(primeiro nome, último nome, sexo, telemóvel, email, data de nascimento, e profissão) encontram-se na posse da câmara municipal num ficheiro CSV.

Contudo, os dados dos habitantes que se encontram neste ficheiro, não se encontram na melhor predisposição e organização, tendo até alguns dados inválidos, assim como alguns habitantes, que não cumprem os parâmetros que são supostos. Devido a estes problemas não é possível por parte da câmara municipal de Famalicão conseguir através destes dados tomar certas decisões informadas, garantido assim o melhor para a sua cidade. Decisões estas como por exemplo obter os habitantes válidos e seus respectivos dados, de uma determinada profissão, de modo a conseguirem contactar essa pessoa para contratar para um determinado projeto ou oportunidade de trabalho.

O objetivo principal da resolução deste problema é a validação e organização dos dados em ficheiros, de diversos tipos, como Excel, XML, JSON e outros, de modo a possibilitar uma visão facilitada, assim como diversas ferramentas para as pessoas responsáveis da câmara municipal pelo tratamentos e análise destes dados tomarem as melhores decisões possíveis.

3. Estratégia Utilizada

No desenvolvimento deste trabalho, foi utilizado a ferramenta Pentaho Data Integration (Kettle) para realizar diversas transformações, assim como jobs nos dados dos habitantes da cidade. Estas operações foram essenciais para garantir que os dados estivessem padronizados, organizados e filtrados, de modo a estarem prontos para utilização pelos interessados.

Padronização e Organização: A primeira etapa consistiu na padronização e organização dos dados, onde as informações provenientes de diferentes fontes foram uniformizadas, eliminando inconsistências e dados inválidos.

Filtragem dos Dados: Na segunda etapa, procedeu-se à, remoção de informações irrelevantes ou redundantes, e filtragem melhorando a qualidade dos dados para as fases seguintes.

Conversão de Dados: A terceira etapa envolveu a conversão entre diversos formatos, como CSV, Excel, XML e JSON, facilitando a integração com diferentes sistemas.

Por fim, todas as transformações visam garantir que os dados estivessem prontos para serem utilizados em dashboards, relatórios e outras ferramentas de análise, assegurando a entrega de resultados precisos e consistentes.

4. Transformações

Foram realizadas 3 transformações de forma a manipular os dados dos habitantes da cidade de Famalicão.

Transformação 1:

Nesta transformação, que usou como base de extração dos dados dos habitantes da cidade um ficheiro CSV, com o nome **people_info.csv**, foi realizado um processo de padronização, assim como verificação e filtração dos dados para verificar se estes eram válidos e não nulos, e de seguida a apresentação final de um ficheiro Excel válido com os dados dos habitantes.

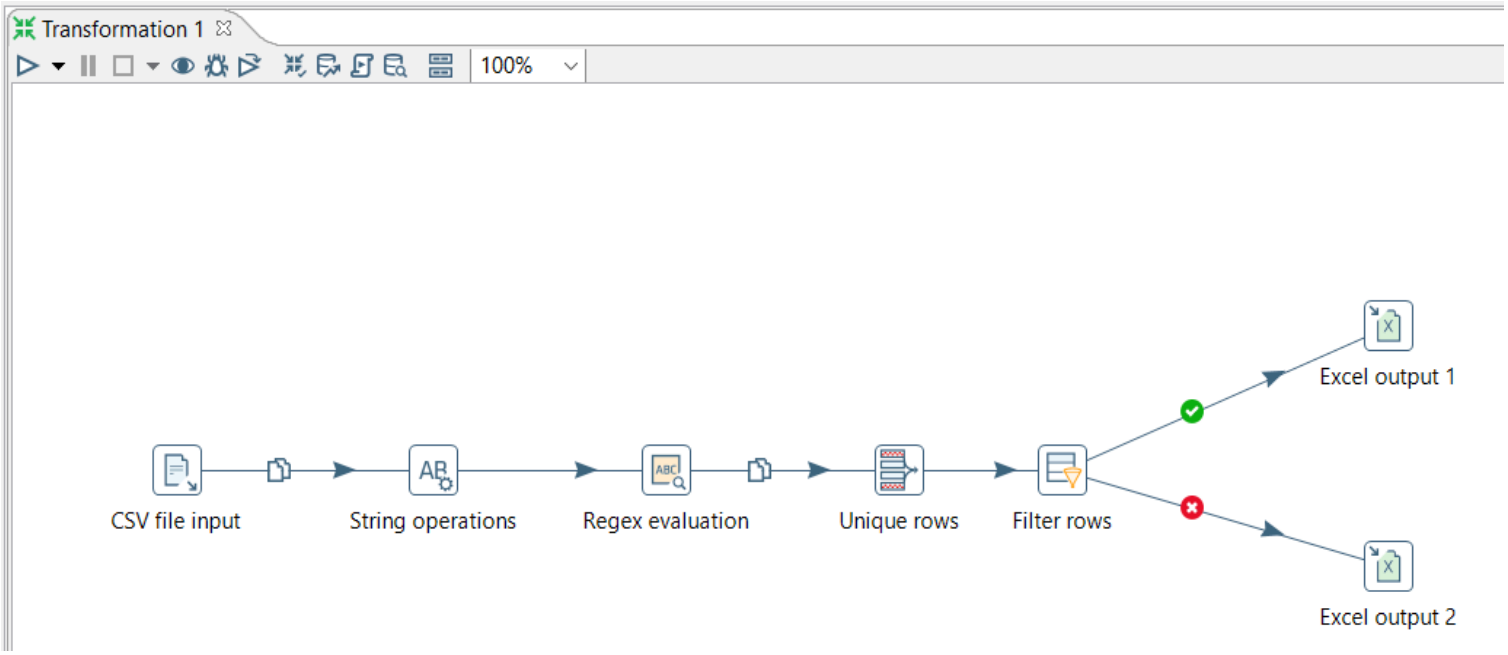


Figura 1 - Transformação 1

1. Leitura de um ficheiro CSV

Primeiramente, foi usado como input um ficheiro CSV, de onde foram extraídos os dados dos habitantes da cidade de Famalicão(primeiro nome, último nome, sexo, telemóvel, email, data de nascimento, e profissão), que se encontra na pasta data/input .

2. Operações com Strings

De seguida com os dado extraídos na etapa anterior foram realizadas as seguintes operações com strings:

- **Remoção de espaços em branco**: Foram removidos os espaços em branco à esquerda e direita das palavras de todos os campos de forma a garantir que não haja dados corrompidos por causa de espaços desnecessários;
- **Padronização de e-mails**: Foram postos em total letra minúscula de forma a padronizar os e-mails e evitar duplicidade de e-mails que seriam diferenciados apenas por maiusculas/minúsculas;
- **Remoção de dígitos**: Para os campos que não devem conter números, como o caso do First_Name, Last_Name, Sex e Job estes foram removidos caso existam.

3. Validação por Regex

De forma a validar os e-mails dos habitantes, estes foram verificados através de regex se tinham o formato correto e se a sua terminação em .net ou .com, com a seguinte expressão regular:

[a-z0-9.\+-]+@[a-z0-9.\+-]+.(com|net)

4. Linhas Únicas

Com o objetivo de não haver habitantes duplicados, foi feita a verificação através do campo Person_ID e Email, a verificação de linhas únicas, isto é, em caso de haver algum habitante duplicado este foi removido.

5. Filtrar Linhas

Depois, foi feita uma filtração por linhas que caso as condições fossem verdade os dados iam para um ficheiro Excel, e caso fossem mentira iam para outro. As condições foram as seguintes:

- **Validation_Email = [Y]** - onde se verifica se o que se encontra neste campo de cada habitante é true, logo o email está validado e no formato correto;
- **[1910-1-01] < Birth_Date < [2024-10-22]** - onde se verifica se o habitante tem uma data de nascimento válida, neste caso se não “nasceu no futuro” ou se tem uma idade superior a 114 anos, o que não é possível;
- Verificação se os campos Person_ID, First_Name, Phone, Birth_Date,e Email não são nulos pois estes são dados obrigatórios.

6. Output em Excel

Por fim, temos dois outputs em Excel resultantes desta transformação, um chamado **people_info.xls** onde ficam os dados validados, assim como organizados e a versão final do ficheiro. O outro ficheiro Excel com o nome **invalid_data.xls** onde ficam os dados dos habitantes que não cumprem os parâmetros devidos, ou tem dados inválidos.

Transformação 2:

Nesta transformação, foi usado o ficheiro que resultou da transformação anterior, **people_info.xls**, com os dados dos habitantes da cidade organizados e padronizados. Foi realizado um processo de definir a categoria de cada pessoa(reformada,criança ou adulto) de acordo com a sua idade, e depois a filtragem destes para ficheiros distintos. Por último, foi feita a filtragem sobre o ficheiro dos adultos para dois novos ficheiros JSON, um que continha os adultos de sexo feminino e outro do sexo masculino.

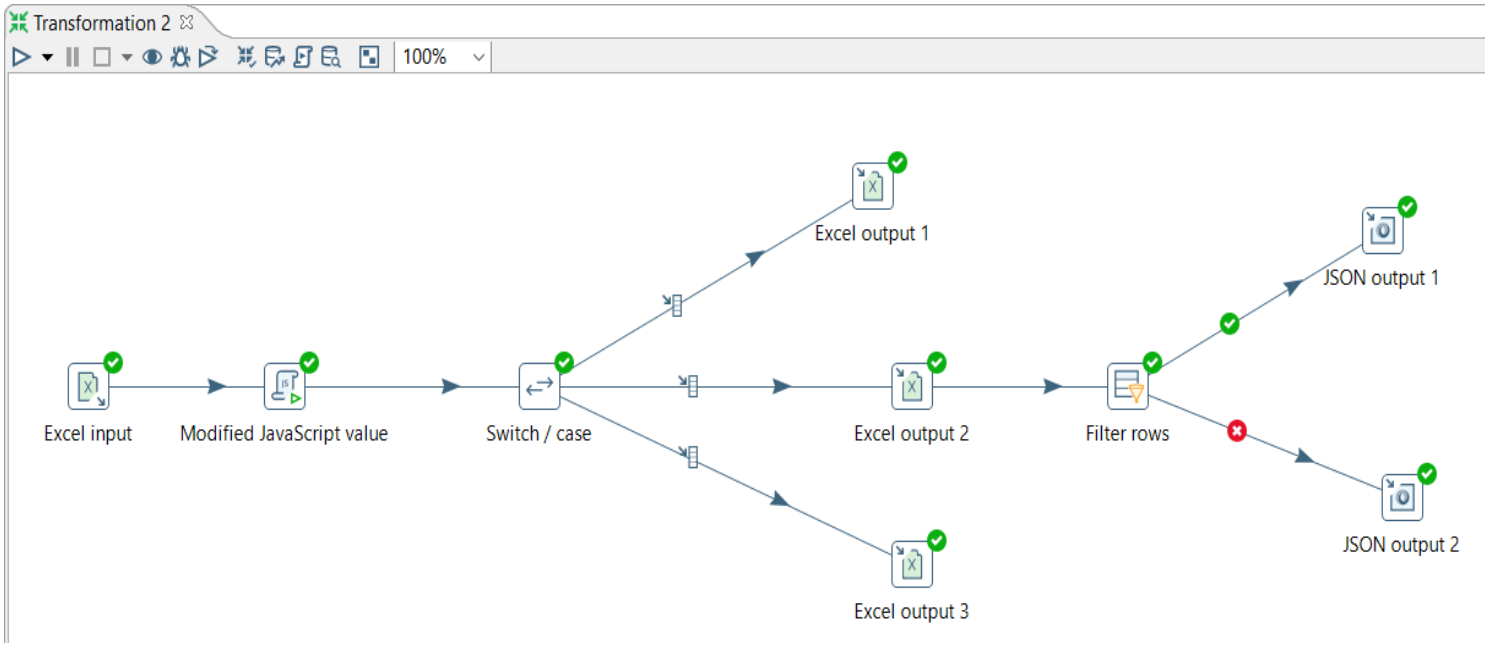


Figura 2 - Transformação 2

1. Leitura de ficheiro Excel

Primeiramente, foi usado como input um ficheiro Excel, de onde foram extraídos os dados dos habitantes da cidade de Famalicão(primeiro nome, último nome, sexo, telemóvel, email, data de nascimento, e profissão), que se encontra na pasta data/output

2. JavaScript

Neste step foi criado um código em Javascript que de acordo com a data de nascimento do habitante, era feito a verificação de se este estava abaixo dos 18 anos e era definido como criança, se este estava entre 18 e 67 anos era definido como adulto eu em caso de maior de 67 anos este ser definido como reformado. O resultado desta verificação ia ser armazenado num novo campo temporário que foi usado no step seguinte.

3. Switch Case

Neste step de acordo com a informação presente no campo criado no step anterior, em que atribuía de acordo com a idade do habitante a opção Criança,Adulto ou Reformado foi redirecionado para três novos ficheiros Excel os dados de determinado habitante.

4. Output em Excel

Do step anterior resultaram como output 3 ficheiros Excel, um que contém os habitantes com idade inferior a 18, outros com os habitantes entre 18 e 67 anos, e outro com os habitantes com mais de 67 anos, com respetivamente os seguintes nomes: **kids_info.xls** , **adults_info.xls**, **reformed.xls** .

5. Filtragem por Linhas

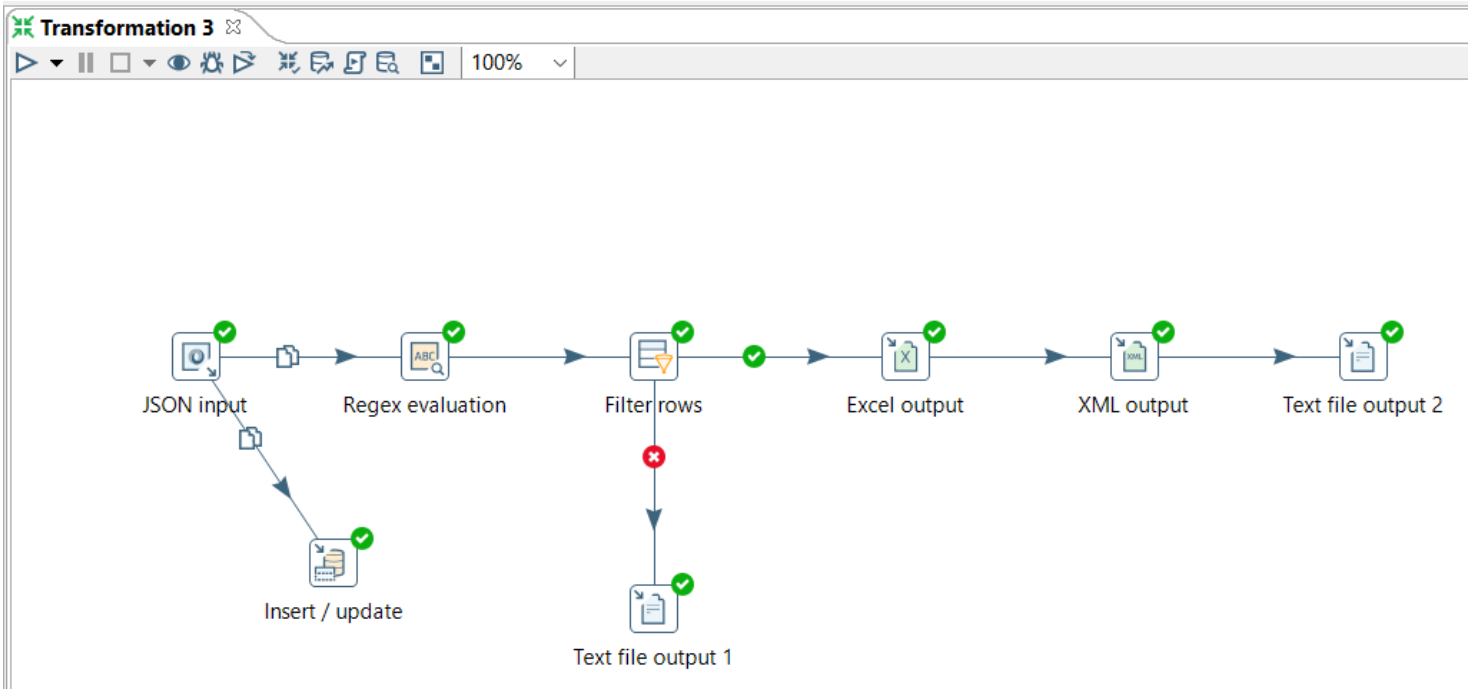
De seguida, foi feita uma filtragem por linhas sobre o ficheiro **adults_info.xls** com o objetivo de separar em dois ficheiros os habitantes. A condição verificada era se o campo Sex tinha nele a string “Male”, e em caso positivo ia para um ficheiro os dados daquele habitante, em caso negativo ia para o outro.

6. Output em JSON

Por último temos dois ficheiros de output no formato JSON, em que um tem o nome de **man_info.json**, que contém os dados dos habitantes adultos do sexo masculino e outro com o nome **female_info.json**, que contém os dados dos habitantes adultos do sexo feminino.

Transformação 3:

Nesta transformação, foi usado o ficheiro JSON que resultou da transformação anterior, com os dados dos habitantes que tem entre 18 e 67 anos de idade, e que são do sexo masculino. De seguida foram inseridos estes dados numa base de dados. Também foi este ficheiro filtrado para num ficheiro novo apresentar só os habitantes que eram engenheiros, e por último foi transformado em diversos tipos de ficheiros como Excel, XML e ficheiro de texto.



Ficheiro 3 - Transformação 3

1. Leitura ficheiro JSON

Primeiro foi obtido os dados do ficheiro JSON, com o nome **male_info.json** que continha os dados dos habitantes do sexo masculino com idade entre 18 e 67 anos(primeiro nome, último nome, sexo, telemóvel, email, data de nascimento, e profissão), que se encontra na pasta data/output

2. a) Inserção/Update numa BD

Os dados extraídos do ficheiro JSON foram inseridos/atualizados na base de dados com o nome **Habitantes**.

2. b) Validação do Regex

Neste step foi realizado uma verificação se no campo Job se encontrava a palavra engineer que ia filtrar os habitantes com profissão na área de engenharia. A expressão regular usada foi a seguinte:

[a-zA-Z]+\s(engineer)

3. Filtrar Linhas

De seguida, foi feita uma filtragem por linhas sobre o ficheiro **male_info.json** com o objetivo de separar em dois ficheiros os habitantes. A condição verificada era se o campo Verification que tinha sido gerado no set anterior tinha nele true(Y) em caso de a pessoa ter como profissão engenharia estes habitantes iam para um ficheiro novo, em caso de falso(N) iam para outro ficheiro.

4. Output em Excel

Temos dois outputs um em Excel e outro em ficheiro de texto resultantes desta transformação, um chamado **engineers.xls** onde ficam os dados habitantes masculinos que são engenheiros. O outro ficheiro de texto com o nome **othersJobs.xls** onde ficam os dados dos habitantes que têm outras profissões.

4. Conversão em XML

Neste step foi feita uma conversão do ficheiro **engineers.xls** que se encontrava no formato Excel para o formato XML, **engineers.xml** .

5. Conversão em Ficheiro de Texto

Neste step foi feita uma conversão do ficheiro **engineers.xml** que se encontrava no formato XML para o formato de ficheiro de texto, **engineers.txt** .

5. Job

Foi realizado um job com o objetivo de executar diversas transformações sequencialmente, que começou com um ficheiro CSV com os dados dos habitantes de Famalicão. Neste job é feita a padronização e organização dos dados no primeiro transform, depois a filtração dos habitantes específicos de uma condição para outros ficheiros, de seguida é realizada a transformação dos ficheiros em outros tipos como XML, JSON, Excel, ficheiro de texto e inclusive inserção dos dados numa base de dados. Por último, deste job resulta um output de diversos ficheiros que vão ser enviados por mail com uma mensagem para um utilizador.

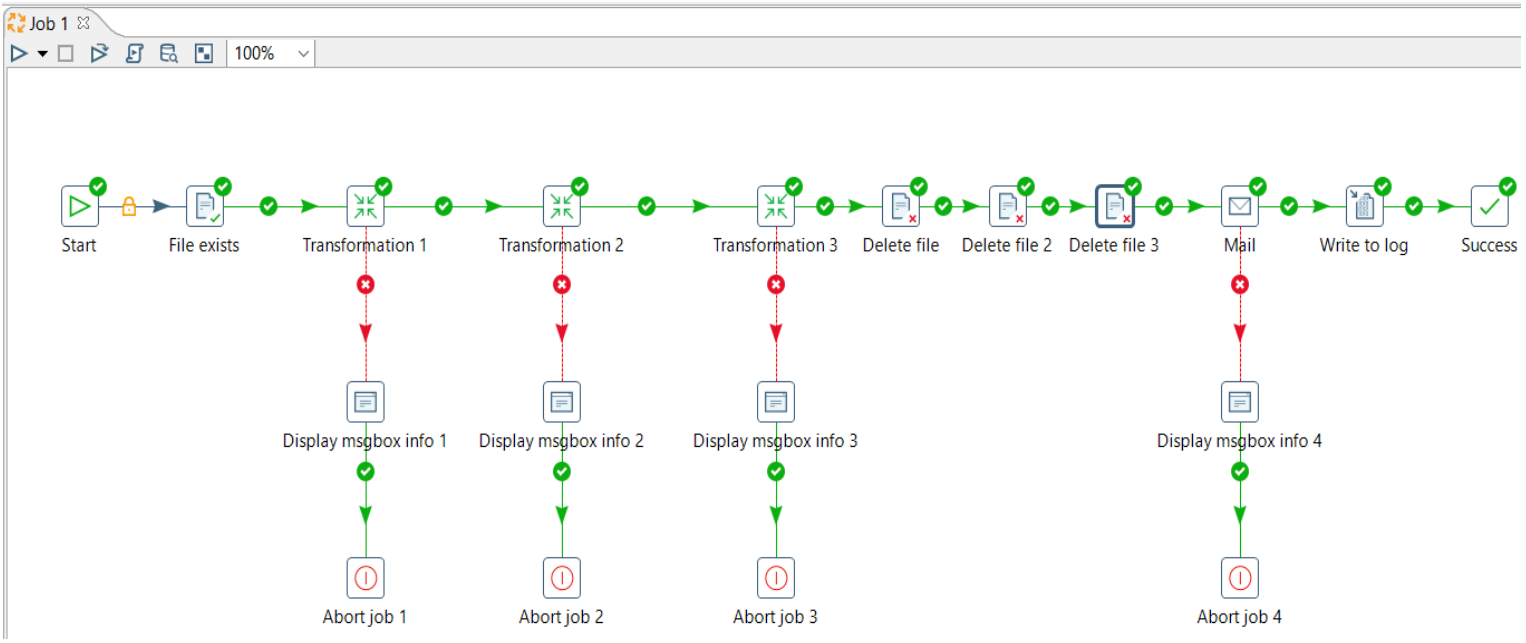


Figura 4 - Job

1- Verificar Ficheiro Existente

Começa o job, e logo após há uma verificação se o ficheiro **people_info.csv** existe, pois este é o que vai ser usado no transform 1 que vai dar origem aos diversos processos e ficheiros consequentes criados e usados noutras transformações.

2 - Transformação 1

Neste step é executada a transformação 1, que tem como input o ficheiro **people_info.csv,** de seguida faz nele todo o processo de padronização e organização dos dados dos habitantes de Famalicão de forma a estes estarem válidos, e tem como output um ficheiro Excel com o nome, **people_info.xls.**

3 - Transformação 2

Neste step é executada a transformação 2, que tem como input o ficheiro **people_info.xls**, que vai sofrer uma filtragem para três novos ficheiros Excel, que um tem os habitantes menores de 18 anos, outro os habitantes entre 18 e 67 anos, e outro maiores de 67. De seguida o ficheiro com os habitantes com idade entre 18 e 67, que tem o nome de **adults_info.xls**, vai sofrer uma filtração para dois ficheiros JSON em que o primeiro vai ter os adultos do sexo masculino e o outro os adultos do sexo feminino, com o nome **male_info.json** e **female_info.json** respetivamente.

4 - Transformação 3

Neste step é executado a transformação 3, que tem como input o ficheiro **male.json** e que de seguida vai inserir/atualizar na base de dados Habitantes estes dados dos respectivos habitantes. Além disso vai através do ficheiro de input verificar quais são os habitantes que trabalham em engenharia, filtrando estes para um ficheiro Excel chamado **engineers_info.xls** que vai ser o ficheiro final. Depois pra finalizar ainda converte este em vários formatos de ficheiro como XML e ficheiro de texto.

5 - Ficheiros eliminados

Neste steps são eliminados vários ficheiros que não têm interesse para serem enviados no mail para o utilizador.

6 - Email

Neste step é enviado um email com uma mensagem para o utilizador indicado, neste caso a26013@gmail.com através do email a26013@gmail.com , com os ficheiros que resultaram dos diversos transforms e que são considerados importantes para o envio para o responsável da câmara municipal de Famalicão, como é o caso do **engineers_info.xls** que vai fornecer a informação de todos os engenheiros do sexo masculino e que não estão reformados, podendo ser contratados para um determinado projeto ou emprego.

5- Sucesso

Finalmente é apresentado no log a informação de sucesso na execução do job, e este é concluído.

6. Demonstração em Vídeo

No link apresentado abaixo, encontra-se um vídeo realizado com o objetivo de demonstrar o funcionamento na totalidade e na perfeição do trabalho desenvolvido para a unidade curricular.

https://www.youtube.com/watch?v=vH6HjCwaAD8&t=16s&ab_channel=PedroRei

7. Conclusão

O presente trabalho demonstrou a importância dos processos de ETL (Extract, Transform, Load) na integração e gestão eficaz de dados, particularmente na perspectiva das organizações que necessitam de informações precisas e estruturadas para a tomada de decisões.

Através da utilização da ferramenta Pentaho Kettle, foi possível explorar as diversas fases envolvidas na criação de um processo de ETL, desde a extração de dados de múltiplas fontes até à transformação e carga em sistemas de destino.

Os desafios enfrentados ao longo deste percurso, como a validação e organização de dados, evidenciam a complexidade envolvida na gestão de grandes volumes de informação. Contudo, as soluções implementadas permitiram não apenas a melhoria da qualidade dos dados, mas também uma visão mais clara e acessível das informações, facilitando o trabalho dos responsáveis pela análise e gestão na Câmara Municipal de Famalicão.

Em suma, a implementação de processos de ETL revelou-se fundamental para a optimização das operações de tratamento de dados, contribuindo significativamente para a eficiência administrativa e a qualidade das decisões tomadas. Este trabalho sublinha a necessidade contínua de adoptar metodologias eficazes para a gestão da informação, permitindo que as organizações respondam de forma adequada e oportuna aos desafios do ambiente actual.

8. Bibliografia/Webliografia

- Documentação fornecida pelo docente da Unidade Curricular;

- https://docs.hitachivantara.com/r/en-us/pentaho-data-integration-and-analytics/9.4.x/mk-95pdia000/getting-started-with-pdi/pentaho-data-integration-pdi-tutorial/step-1-extract-and-load-data/edit-and-save-the-transformation