Trabalho prático 1
Integração de Sistemas de Informação

Docente: Luís Ferreira

Realizado por:

Gonçalo Gomes (a25455)
Hugo Monteiro (a27993)

Licenciatura em Engenharia Sistemas Informáticos
3º ano

Índice

Índice de figuras

Introdução

No contexto atual de crescente digitalização e transformação dos negócios, a integração eficaz de sistemas de informação tornou-se um fator estratégico para o sucesso organizacional. Processos de integração que garantam a correta extração, transformação e carregamento de dados (ETL) são fundamentais para permitir que empresas mantenham a coesão e o valor dos dados entre diferentes sistemas, ferramentas e plataformas.
Este trabalho tem como objetivo consolidar o conhecimento sobre integração de sistemas de informação a nível de dados, explorar novas ferramentas e tecnologias, e desenvolver as capacidades técnicas na construção de soluções ETL, contribuindo assim para a melhoria da assimilação de conceitos e metodologias apresentadas ao longo da unidade curricular.

Contextualização

O tema escolhido foi a obtenção de vários tipos de vinhos, com certas características de uma API pública online, utilizando a plataforma KNIME e Pentaho Data Integration (Kettle).

Nestes processos, pode-se verificar a obtenção de dados através de uma API externa, seguida da criação e respetiva configuração de ficheiros necessários ao fluxo de trabalho. O pipeline implementa operações de filtragem, seleção, ordenação e eliminação de dados, assegurando a qualidade e padronização da informação processada.
Durante o processamento, são realizadas operações de importação e exportação de dados entre ficheiros em formatos CSV, JSON, XML e XLS, garantindo compatibilidade com sistemas de origem e destino. O sistema incorpora a utilização de expressões regulares para validação e normalização de campos textuais, permitindo extrair e filtrar informações com precisão.
Para suportar a comunicação operacional, o processo inclui o envio de emails com resultados.
Por fim, é possível comprovar a geração automática de logs pelo próprio KNIME ou logs retirados da base de dados que são enviados pelo Pentaho, assegurando diagnóstico de falhas e documentação completa do histórico de execuções do pipeline ETL implementado.

Workflows - Knime

GetWinesFromAPI

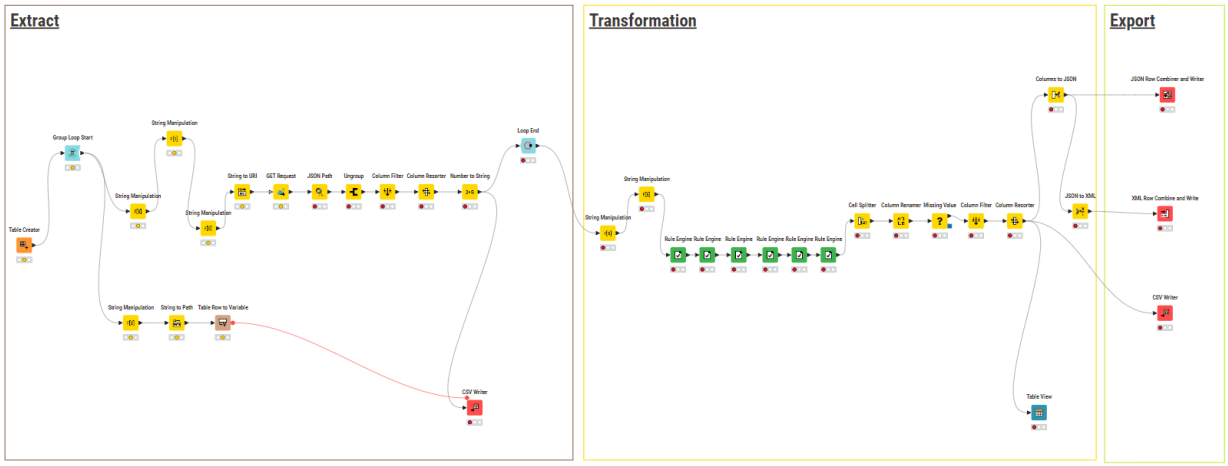


Figura 1 GetWinesFromAPI

Na imagem acima, é criada uma tabela de controlo com duas colunas, url e winetype, que lista os endpoints a consultar, esta tabela alimenta um loop que percorre cada linha e, para cada endpoint, constrói o URI, efetua o pedido à API e extrai o conteúdo JSON para uma estrutura tabular única, permitindo consolidar todos os registos num único dataset para as etapas seguintes do fluxo. Durante o próprio loop, para garantir persistência incremental e facilitar depuração, cada porção de dados obtida de cada endpoint é também escrita imediatamente em ficheiros CSV individuais, criando assim um histórico por chamada enquanto a consolidação continua.

Concluída a fase de extração, entra-se na etapa de transformação: aplicam-se dois nós de String Manipulation para limpeza textual, removendo quebras de linha (\n) de campos como country e region e suprimindo o sufixo “ratings” nas reviews, de modo a padronizar os valores e melhorar a legibilidade e a consistência. Em seguida, uma sequência de nós Rule Engine valida a completude por coluna; sempre que é detetado um campo vazio, o valor é substituído por “UNKNOW”, assegurando que não permanecem células em branco e que os consumidores a jusante têm um marcador explícito de dado em falta. Após a normalização e verificação, os dados resultantes são exportados para três formatos paralelos — JSON, XML e CSV.

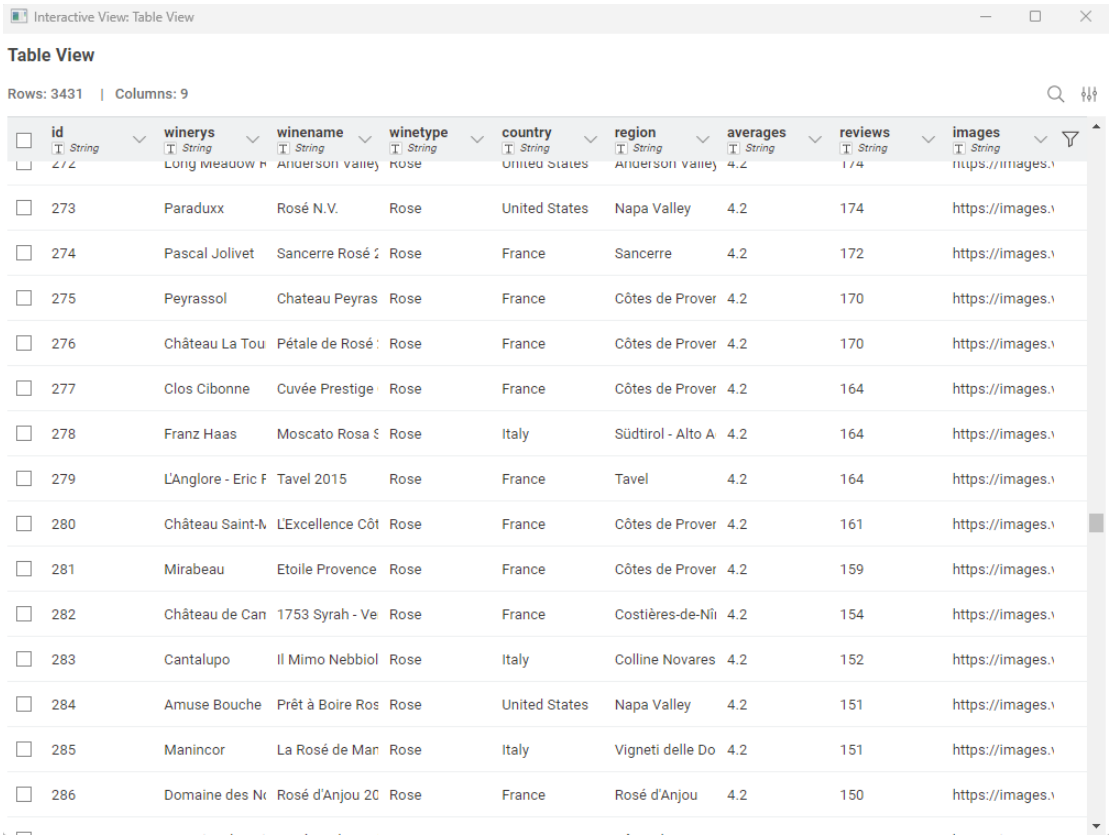


Figura 2 Tabela de todos os vinhos

FilteredDataset

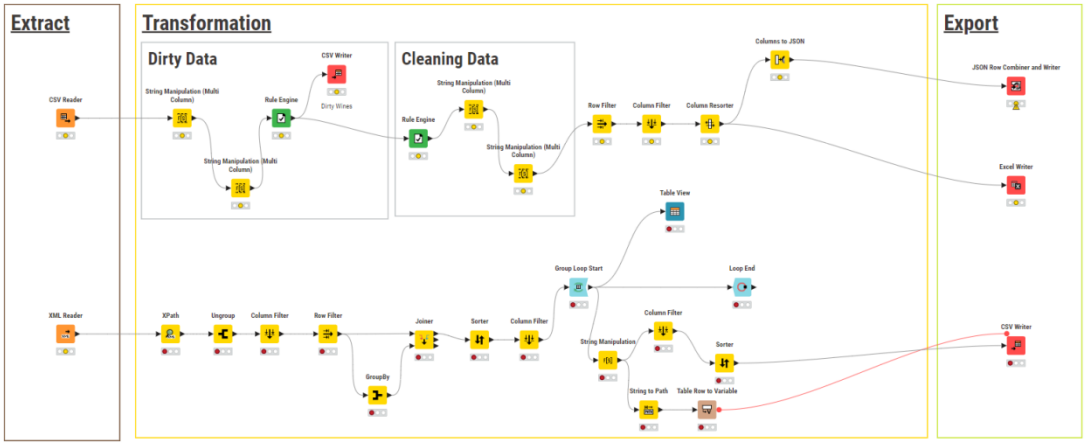


Figura 3 FilteredDataset

O workflow apresentado acima está organizado em três grandes etapas: **Extract**, **Transformation** e **Export**.

No fluxo de cima, começo por carregar um ficheiro CSV com dados inicialmente bem formatados, provenientes de outro workflow. De seguida, de forma intencional, altero e desformato esses dados, introduzindo erros e inconsistências através de várias manipulações e do uso do Rule Engine, simulando situações reais de dados sujos. Estes dados "desformatados" são então guardados num novo ficheiro CSV para documentar o estado intermédio dos dados. Logo depois, aplico novamente operações de limpeza e normalização para restaurar os dados ao seu formato correto, recorrendo a nós de String Manipulation (Multi Column) e Rule Engine para padronização, remoção de caracteres indesejados e tratamento de campos em falta. Com os dados limpos, aplico uma filtragem rigorosa para selecionar apenas os vinhos tintos portugueses da região do Douro, que tenham mais de 300 reviews, restringindo assim o conjunto aos registos relevantes para análise.

No fluxo de baixo, o objetivo é identificar, para cada tipo de vinho (por exemplo, tinto, branco, etc.), aquele que possui o maior número de reviews em cada um dos países: Portugal, França, Itália e Espanha. Para isso, são efetuados filtros e agrupamentos sucessivos com recurso aos nós GroupBy, Sorter e filtros de colunas, garantindo que, para cada combinação de país e tipo de vinho, apenas o vinho mais popular em termos de reviews é selecionado. O processo segue com a preparação dos dados, passagem para variáveis e escrita dos resultados finais em múltiplos formatos—CSV, Excel e JSON—permitindo fácil integração e consulta posterior.

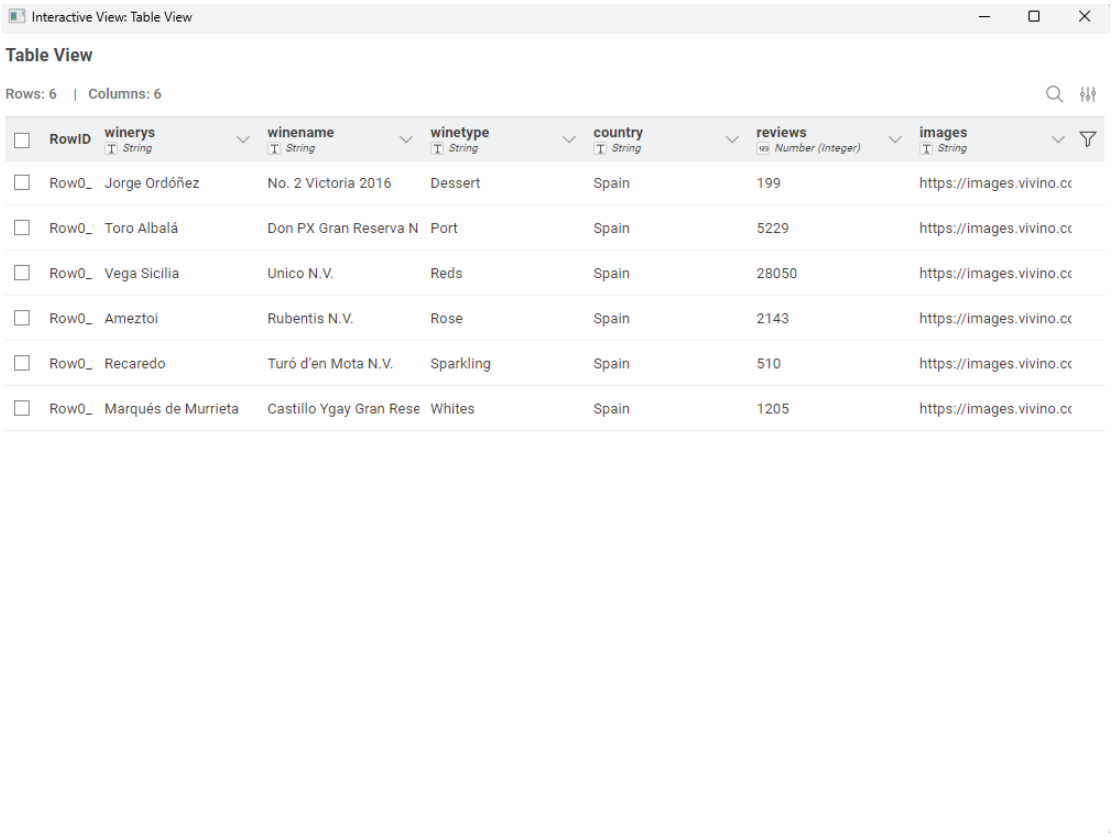


Figura 4 Table dos vinhos espanhóis com mais reviews de cada tipo de vinho

SendEmail

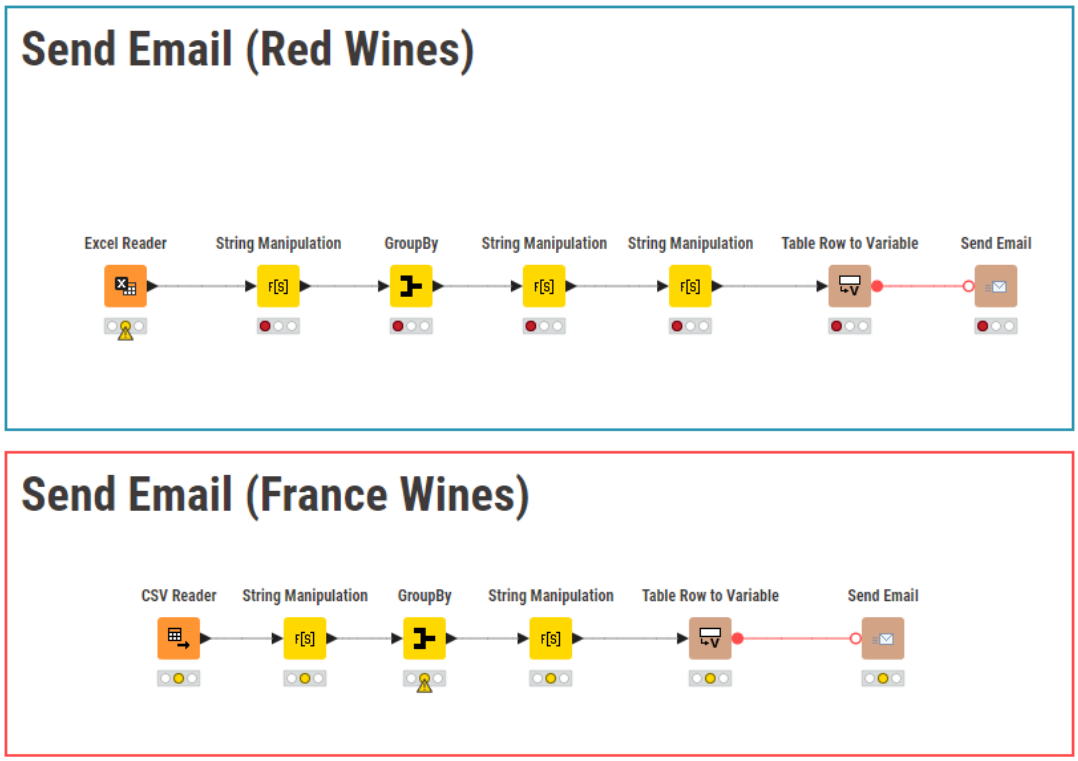


Figura 5 SendEmail

Na figura 5, podemos ver neste workflow , após ler o ficheiro de resultados, a tabela é convertida para um corpo de email em HTML recorrendo a uma cadeia de nós String Manipulation: os primeiros compõem o markup da tabela
(tags <table>, <thead>, <tbody>, <tr> e <td>) e inserem dinamicamente os cabeçalhos e linhas a partir dos valores das colunas, enquanto o último acrescenta um bloco de CSS inline para estilização (larguras, bordas, alinhamento, cores e tipografia), produzindo um HTML limpo e responsivo.
Esta abordagem foi necessária porque, para o contexto deste envio, não está a ser usado um nó nativo que exporte automaticamente a tabela para HTML, pelo que a conversão e o embelezamento são construídos via manipulação de strings antes de alimentar o nó de envio de email.

Job

ETL_Job

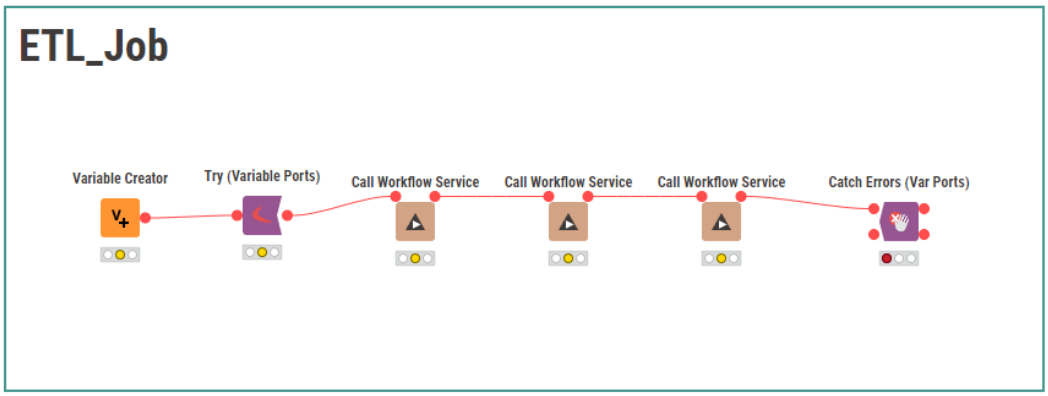
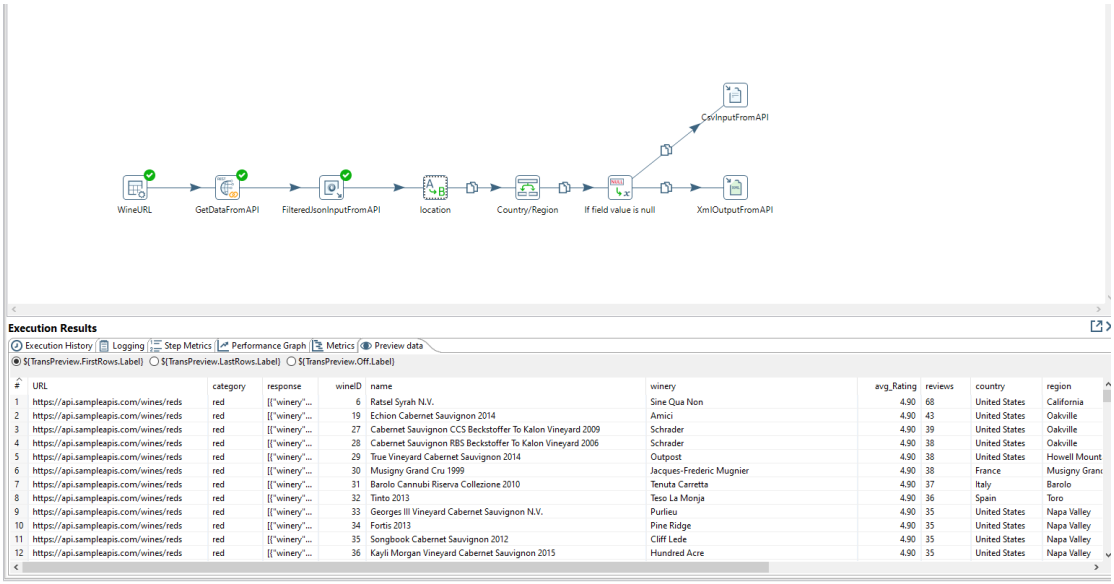


Figura 6 ETL_Job

Este workflow é o orquestrador do ETL: define variáveis de controlo, executa sequencialmente os sub-workflows e captura falhas num único sítio. Primeiro, cria as variáveis globais do job (parâmetros, caminhos, datas e flags) e, dentro de um bloco de tentativa, chama cada workflow de extração, transformação e exportação através de nós de invocação, passando as mesmas variáveis entre etapas para manter consistência. No fim, um bloco de captura trata qualquer erro devolvido, permitindo registar logs, acionar notificações ou marcar o job como falhado para quem agenda a execução.
É, portanto, a camada de orquestração: parametriza, encadeia e protege a execução ponta-a-ponta.

Transformações – Pentaho

GetDataFromApi



#	URL	category	response	wineID	name	winery	avg_Rating	reviews	country	region
1	https://api.sampleapis.com/wines/reds	red	[{"winery"...	6	Ratsel Syrah N.V.	Sine Qua Non	4.90	68	United States	California
2	https://api.sampleapis.com/wines/reds	red	[{"winery"...	19	Echion Cabernet Sauvignon 2014	Amici	4.90	43	United States	Oakville
3	https://api.sampleapis.com/wines/reds	red	[{"winery"...	27	Cabernet Sauvignon CCS Beckstoffer To Kalon Vineyard 2009	Schrader	4.90	39	United States	Oakville
4	https://api.sampleapis.com/wines/reds	red	[{"winery"...	28	Cabernet Sauvignon RBS Beckstoffer To Kalon Vineyard 2006	Schrader	4.90	38	United States	Oakville
5	https://api.sampleapis.com/wines/reds	red	[{"winery"...	29	True Vineyard Cabernet Sauvignon 2014	Outpost	4.90	38	United States	Howell Mount
6	https://api.sampleapis.com/wines/reds	red	[{"winery"...	30	Musigny Grand Cru 1999	Jacques-Frederic Mugnier	4.90	38	France	Musigny Gran
7	https://api.sampleapis.com/wines/reds	red	[{"winery"...	31	Barolo Cannubi Riserva Collezione 2010	Tenuta Carretta	4.90	37	Italy	Barolo
8	https://api.sampleapis.com/wines/reds	red	[{"winery"...	32	Tinto 2013	Teso La Monja	4.90	36	Spain	Toro
9	https://api.sampleapis.com/wines/reds	red	[{"winery"...	33	Georges III Vineyard Cabernet Sauvignon N.V.	Purlieu	4.90	35	United States	Napa Valley
10	https://api.sampleapis.com/wines/reds	red	[{"winery"...	34	Fortis 2013	Pine Ridge	4.90	35	United States	Napa Valley
11	https://api.sampleapis.com/wines/reds	red	[{"winery"...	35	Songbook Cabernet Sauvignon 2012	Cliff Lede	4.90	35	United States	Napa Valley
12	https://api.sampleapis.com/wines/reds	red	[{"winery"...	36	Kayli Morgan Vineyard Cabernet Sauvignon 2015	Hundred Acre	4.90	35	United States	Napa Valley

Figura 7 GetDataFromApi

Como podemos ver na figura 7, é realizada a transformação que consiste em extrair os dados de uma API pública de vinhos, processá-los e exportá-los para ficheiros nos formatos **CSV** e **XLM**, o fluxo segue uma abordagem de **ETL** (Extract, Transformation, Load).
A estrutura começa pela extração de dados no **WineURL** onde são definidos todos os URLs da API externa, logo em seguida fazemos a recuperação dos dados em formato **JSON** para serem convertidos em formato tabular **(GetDataFromAPI – FilteredJsonInputFromAPi)**.
Após a conversão podemos iniciar o processo de tratamento dos dados nos passos seguintes (**location – If field value is null).** Por fim os dados são guardados em **XML** e **CSV** permitindo a sua utilização em transformações posteriores**.**

FilteredDataFromApi

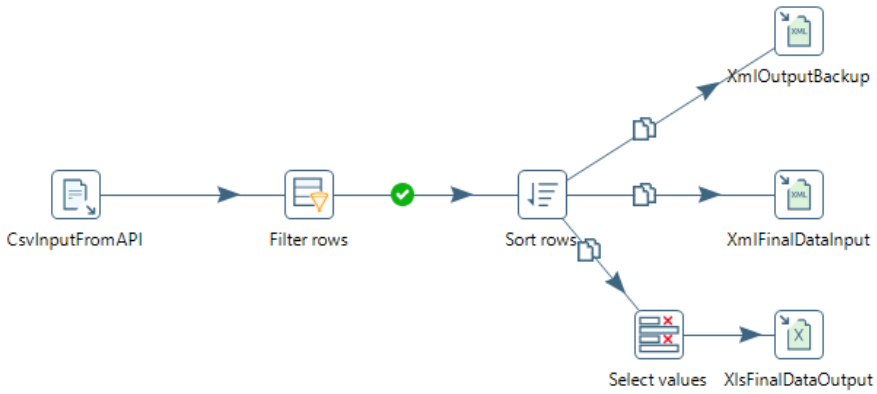


Figura 8 FilteredDataFromAPI

Nesta fase do processo (figura 8), é realizada a filtragem e organização dos dados extraídos da API, garantindo que apenas os dados relevantes são mantidos para a análise final. A transformação tem início no **CsvInputFromAPI,** que lê os dados exportados da **GetDataFromAPI,** transformação anteriormente falada.
Em seguida, é feita a filtragem **(Filter rows)** para garantir que somente os dados necessários são utilizados, sendo estes os vinhos tintos portugueses da região do Douro com mais de 300 reviews todos ordenados de forma decrescente **(Sort rows)**. Por fim a informação é guardada em 3 ficheiros diferentes, o primeiro cria uma cópia de segurança no **XmlOutputBackup**, o **XmlFinalDataInput** para uso posterior e o último guarda em excel para ser fácil de analisar os resultados **XlsFinalDataOutput.**

SaveHtml

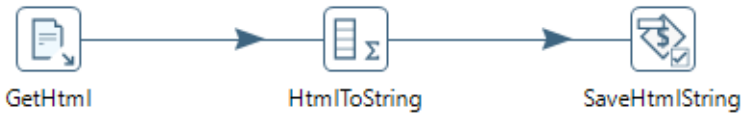


Figura 9 SaveHtml

Na figura 9, é realizado o processamento de um ficheiro **HTML**, com o objetivo de o preparar para utilizar o conteúdo de forma dinâmico no envio de emails. A transformação inicia no **GetHtml** onde é feita a leitura de um ficheiro HTML que é previamente criado e guardado. Em seguida o HTML é transformado numa string **(HtmlToString)** para ser guardado numa variável tornando-o acessível noutras transformações ou jobs dentro do projeto (**SaveHtmlString).**

SendEmail

Figura 10 SendEmail

Na transformação **SendEmail (figura 10)**, é implementado o processo de envio automático de emails, com base nos resultados previamente obtidos e nos utilizadores registados na base de dados. O fluxo inicia-se no **GetUser**, onde são recolhidos os dados dos destinatários, como o nome, email e estado de atividade a partir daí é feita uma validação dos emails dos mesmos onde é verificado se os emails guardados estão bem estruturados **(email validation).**

Dos emails válidos verifica-se a sua atividade, pois apenas os ativos são relevantes para receberem o email **(Filter rows)**. Por fim no **EmailContent** realiza-se a construção do email recorrendo ao HTML previamente preparado, neste ponto é definido o corpo do email, que inclui o título, a saudação e o HTML, logo em seguida o email é enviado para os utilizadores válidos a partir do **sendEmail.**

GetLogs

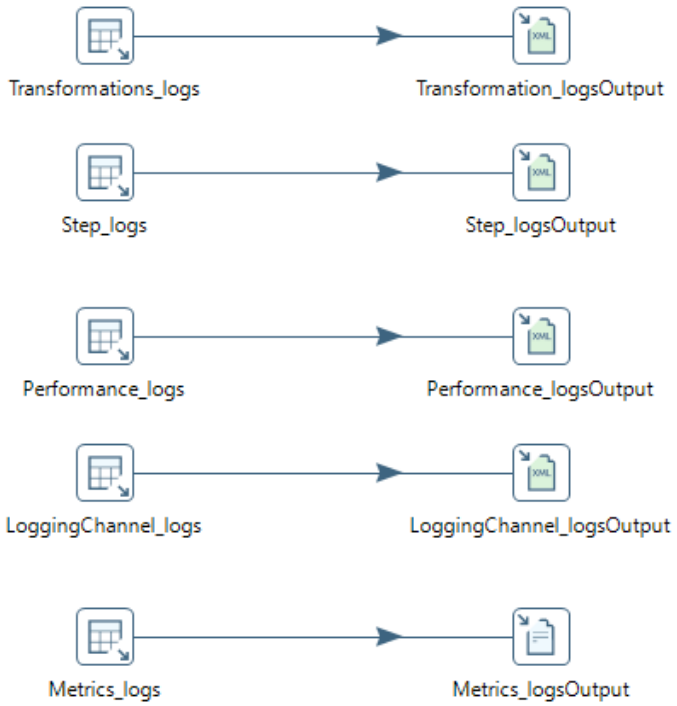


Figura 11 GetLogs

Na figura 11, é realizada a obtenção de dados de uma base de dados, de 5 tabelas diferentes, todas relacionadas a logs diferentes. Cada tabela vai dar origem a um ficheiro xml com exceção do **Metrics_logs** que é guardado num ficheiro texto.

Base de dados

```
create table LoggingChannel_logs(
ID_BATCH int,
CHANNEL_ID varchar(255),
LOG_DATE datetime,
LOGGING_OBJECT_TYPE varchar(255),
[OBJECT_NAME] varchar(255),
OBJECT_COPY varchar(255),
REPOSITORY_DIRECTORY varchar(255),
[FILENAME] varchar(255),
[OBJECT_ID] varchar(255),
OBJECT_REVISION varchar(255),
PARENT_CHANNEL_ID varchar(255),
ROOT_CHANNEL_ID varchar(255)
);

create table Metrics_logs(
ID_BATCH int NOT NULL,
CHANNEL_ID varchar(255),
LOG_DATE datetime,
METRICS_DATE datetime,
METRICS_CODE varchar(255),
METRICS_DESCRIPTION varchar(255),
METRICS_SUBJECT varchar(255),
METRICS_TYPE varchar(255),
METRICS_VALUE varchar(255)
);

create table Utilizadores(
IdUtilizador int IDENTITY(1,1) primary key,
Nome varchar(255) NOT NULL,
Email varchar(255) NOT NULL,
Contacto varchar(255) NOT NULL,
Pais varchar(255) NOT NULL,
Ativo bit NOT NULL DEFAULT 1
);

INSERT INTO Utilizadores(Nome, Email, Contacto, Pais, Ativo)
VALUES
('Ana Ferreira', 'ana.ferreira@gmail.com', '912345678', 'Portugal', 0),
('João Martins', 'joao.martins@hotmail.com', '934567890', 'France', 0),
('Maria Oliveira', 'maria.oliveira@gmail.com', '965432198', 'Italy', 0),
('Pedro Sousa', 'pedro.sousa@email.com', '918765432', 'Spain', 0),
('Inês Rocha', 'ines.rocha@email.com', '937654321', 'United States', 0),
('Ricardo Gomes', 'ricardo.gomes@gmail.com', '915678234', 'Argentina', 0),
('Carla Ribeiro', 'carla.ribeiro@hotmail.com', '963482716', 'Portugal', 0),
('Tiago Monteiro', 'tiago.monteiro@gmail.com', '928374651', 'France', 0),
('Sofia Nunes', 'sofia.nunes@email.com', '914726391', 'Italy', 0),
('Miguel Costa', 'miguel.costa@hotmail.com', '931847265', 'Spain', 0),
('Hugo Monteiro', 'hugo-_-monteiro@hotmail.com', '123456789', 'Italy', 1);
```

Figura 12 Excerto da base de dados local

Na figura 12, mostra um excerto da base de dados local onde todas as transformações do projeto estão integradas, esta desempenha um papel fundamental na gestão de utilizadores e no registo de logs. Embora a maioria das transformações trabalhe diretamente com dados provenientes da API, a base de dados é essencial para a fase final do processo.

Project

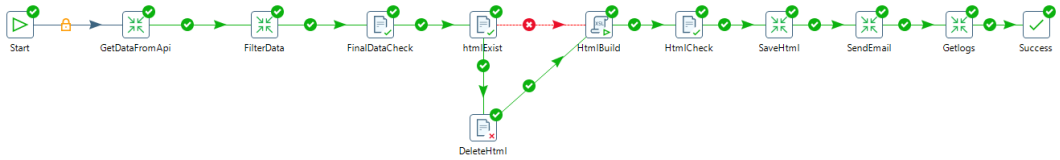


Figura 13 Project

A figura 13 representa o **job principal**, responsável por juntar todas as transformações do projeto, a execução é sequencial e controlada. O job inicia-se com a extração dos dados e após o processamento e filtragem, valida a existência dos ficheiros finais. A partir desse ponto, implementa uma condição para com o HTML, onde caso exista uma versão anterior essa é eliminada de forma a não haver informação duplicada e desatualizada, caso não exista um ficheiro HTML esse mesmo é criado. Por fim o job termina com o envio do email.

Demonstração – KNIME

Demonstração-Pentaho

Conclusão

Neste trabalho conseguimos consolidar e aprofundar os nossos conhecimentos sobre os processos ETL, e sobre o que é a integração de dados e as suas aplicações. Aprendemos também diferentes tipos de ficheiros e a melhor maneira para usufruir deles. Com a utilização de diferentes ferramentas, KNIME e Pentaho, repara-mos que o Pentaho é mais simples e direto mas em contra partida o KNIME é um pouco mais difícil mas bem mais apresentável.

Bibliografia

https://pentaho.com/products/pentaho-data-integration/

https://sampleapis.com/api-list/wines

https://regex101.com/

https://github.com/HugoMonteiro04/TP1-ISI

https://github.com/GoncaloGomes10/TP01-ISI-KNIME.git

https://www.youtube.com/@KNIMETV