

Relatório do trabalho da disciplina de ISI

Processos ETL

Carlos Oliveira - 20530

Licenciatura em Engenharia de Sistemas Informáticos

Docente:

Luís Ferreira

Outubro de 2024

Afirmo por minha honra que não recebi qualquer apoio não autorizado na realização deste trabalho prático. Afirmo igualmente que não copiei qualquer material de livro, artigo, documento web ou de qualquer outra fonte exceto onde a origem estiver expressamente citada.

Carlos Oliveira - 20530

Índice

<i>Enquadramento</i>	1
<i>Problema</i>	2
<i>Estratégia Utilizada</i>	3
<i>Criação de Dados fictícios</i>	4
<i>Criação de uma Base de Dados</i>	4
<i>Transformações</i>	5
<i>Jobs</i>	6
1º Job – Carregamento, Transformação e Exportação de dados para uma Base de Dados	6
Leitura de Ficheiro e Tratamento de campos vazios	6
Filtragem e Segmentação de Dados	8
Transformação de Dados	10
Inserção e Exportação de Dados	11
Filtragem e Reorganização de Colunas	13
Exportação Final	14
2º Job - Conversão de CSV para XML:	15
Filtragem de Colunas:	15
Conversão de Colunas Para XML	16
Combinação de Colunas	17
Escrita do Ficheiro XML	18
3º Job - Envio de Tabela Por Email	19
Leitura do Ficheiro CSV	19
Conversão de Tabela para HTML	20
Envio do Email	20
4º Job – Get Request a uma API de Coordenadas	23
Extração de Dados	23
Conversão de Json para Tabela	23
Conversão de colunas em linhas e identificação destas	24
Junção das Colunas Previamente Identificadas e filtragem das Colunas Desnecessárias	25
Exportação do Ficheiro CSV Resultante	26
<i>Conclusão</i>	27
<i>Referências Bibliográficas</i>	27
<i>Código QR do Vídeo de Demonstração</i>	28

Lista de Figuras

Figura 1 - Criação de Dados Fictícios	4
Figura 2 - Criação de uma Base de Dados	4
Figura 3 - 1º Job	6
Figura 4 - Configuração de Missing Value	6
Figura 5 - Dados Vazios	7
Figura 6 - Preenchimento dos Dados Vazios	7
Figura 7 - Configuração de Number to String	7
Figura 8 - Conversão de Inteiros para String	8
Figura 9 - Filtragem de Dados por Idade	8
Figura 10 - Filtragem por Género	9
Figura 11 - Filtragem por Referência	9
Figura 12 - Filtragem por Género e Idade	9
Figura 13 - Configuração de String Splitter	10
Figura 14 - Conexão à base de dados	11
Figura 15 - Criação da Tabela "ISI"	11
Figura 16 - Criação das tabelas Necessárias	12
Figura 17 - Configuração de DB Insert	12
Figura 18 - Base de Dados Preenchida	13
Figura 19 - Filtragem de Tabelas Desnecessárias	13
Figura 20 - Ordenação de Colunas	14
Figura 21 - Exportação do Ficheiro CSV	14
Figura 22 - Processo de Conversão de CSV para XML	15
Figura 23 - Column Filter	15
Figura 24 - Tabela Resultante de Column Filter	15
Figura 25 - Configuração de Column to XML Para Género	16
Figura 26 - Configuração de Column to XML Para First Name	16
Figura 27 - Tabela Resultante da Coversão de Gender para XML	16

Figura 28 - Tabela Resultante da Coversão de First Name para XML	16
Figura 29 - Configuração de Column Combiner	17
Figura 30 - Configuração de Row Combiner	17
Figura 31 - Resultado de Column Combiner	18
Figura 32 - Resultado de Row Combiner	18
Figura 33 - Configuração do XML Witter	18
Figura 34 - XML Resultante da Conversao de CSV	19
Figura 35 - Envio de Tabela Por Email	19
Figura 36 - Configuração de Table to HTML	20
Figura 37 - Resultado da Conversão para HTML	20
Figura 38 - Configuração do Remetente	21
Figura 39 - Configuração do Cliente SMPT	21
Figura 40 - Criação de uma Flow Variable	22
Figura 41 - Email Recebido	22
Figura 42 - Get Request a API	23
Figura 43 - URL do API no Get Request	23
Figura 44 - Tabela Resultante da Conversao de um Json	24
Figura 45 - Unpivot das Colunas "Name"	24
Figura 46 - Tabela Resultante do Unpivot	24
Figura 47 - Configuração do RowId	25
Figura 48 - Tabela Resultante do RowId	25
Figura 49 - Configuração do Joiner na Filtragem de dois Documentos	26
Figura 50 - Tabela Resultante da Filtragem de Colunas Desnecessárias	26
Figura 51 - Ficheiro CSV Exportado	26
Figura 52 - Código QR do Vídeo de Demonstração	28

Enquadramento

Este Projeto consiste na criação de um fluxo de trabalho automatizado para a manipulação, transformação e integração de dados. Este processo baseia-se em técnicas de ETL (Extraction, Transformation, Load) para converter dados entre vários formatos (como CSV, XML e JSON), extrair informações de APIs externas e enviar relatórios por e-mail, garantindo uma comunicação eficiente e organizada.

No panorama tecnológico atual, as organizações enfrentam o desafio de lidar com uma vasta quantidade de dados dispersos em múltiplos sistemas e formatos, seja para fins de análise, integração de sistemas ou comunicação. O uso de ferramentas como o **Knime** permite a criação de fluxos de trabalho automáticos e modulares garantindo uma maior eficiência em processos repetitivos como extração de dados, conversão dos mesmos, envio de relatórios por email, etc.

Existe uma necessidade crescente das organizações em processar grandes volumes de dados, muitas vezes provenientes de fontes distintas e em formatos diversificados. Com o crescimento da digitalização e da complexidade dos sistemas de informação, torna-se essencial dispor de ferramentas que permitam a automação destes processos, assegurando que os dados são transformados, integrados e distribuídos corretamente.

Problema

O objetivo deste fluxo de trabalho é processar um conjunto de dados e transformá-lo adequadamente para a inserção numa base de dados MySQL e exportação para um ficheiro CSV. O problema concentra-se em lidar com valores ausentes, conversão de tipos de dados, filtragem de linhas com base em critérios específicos, e renomeação de colunas para garantir que os dados estejam no formato adequado para as operações seguintes.

Para além destas operações principais, foram realizados testes adicionais que demonstram a flexibilidade do fluxo de trabalho. Um dos testes envolveu a conversão de um ficheiro CSV para os formatos XML e JSON, comprovando a capacidade do sistema de exportar dados para formatos amplamente utilizados em integrações com outras aplicações e sistemas. Também foi automatizado o envio de uma tabela por e-mail, onde os dados foram convertidos numa tabela em HTML e enviados, facilitando a distribuição de informações para os utilizadores finais. Além disso, foi realizada uma requisição GET a uma API, permitindo a integração de dados de fontes externas. Os dados obtidos em formato JSON foram processados e convertidos numa tabela, tornando-os prontos para análise ou para outras utilizações.

Em resumo, o problema abordado por este fluxo de trabalho reside na necessidade de garantir a qualidade, integridade e flexibilidade dos dados ao longo de todas as fases do processo, desde a sua manipulação inicial até à inserção numa base de dados e à sua distribuição em diferentes formatos. O fluxo foi desenhado para enfrentar estes desafios de forma automatizada e eficiente, assegurando que os dados estão adequadamente preparados para cada uma das operações exigidas em cada etapa do processo.

Estratégia Utilizada

A estratégia utilizada neste fluxo de trabalho foi desenhada para abordar de forma eficiente os desafios associados à preparação e transformação de dados. O processo começa com o tratamento de valores ausentes, onde se identifica e resolve a presença de dados incompletos ou nulos. Esta etapa é essencial para garantir a integridade e consistência da base de dados, uma vez que valores em falta podem comprometer tanto a análise como a utilização dos dados em fases posteriores.

De seguida, a estratégia inclui a conversão de tipos de dados, que envolve transformar certos valores, como números, em texto ou outros formatos adequados. Esta conversão assegura que os dados estão alinhados com os requisitos das aplicações ou da base de dados, sendo um passo fundamental para garantir que a exportação e a inserção dos dados sejam feitas de acordo com os padrões necessários.

Outro aspeto crítico desta estratégia é a filtragem de linhas com base em critérios específicos. Através deste processo, são selecionados apenas os registos que cumprem os requisitos definidos, e excluídos os que não são relevantes ou apropriados para os próximos passos. Esta filtragem assegura que apenas os dados necessários são processados e exportados, evitando ruído desnecessário no conjunto de dados final.

Por último, foi realizada a renomeação de colunas, uma etapa importante para garantir que os nomes dos campos estejam devidamente formatados e alinhados com o padrão esperado. Isto facilita a manipulação e identificação dos dados ao longo das diferentes operações, tanto para a inserção na base de dados como para a exportação para ficheiros ou outros sistemas.

Criação de Dados fictícios

Para a criação de dados fictícios, utilizei a ferramenta Mockaroo, que permite gerar dados sintéticos de forma rápida e eficiente. Assim, foi possível personalizar a estrutura dos dados, definir colunas, tipos de dados e formatos específicos, o que torna os dados gerados adequados para testes, e simulações.

The screenshot shows the Mockaroo interface with the following configuration:

Field Name	Type	Options
id	Row Number	blank: 0 %
name	Full Name	blank: 0 %
age	Number	min: 1 max: 60 decimals: 0 blank: 10 %
email	Email Address	blank: 0 %
gender	Gender	blank: 0 %
job	Job Title	blank: 0 %
country	Country	restrict countries... blank: 0 %

Buttons: + ADD ANOTHER FIELD, GENERATE FIELDS USING AI...

Rows: 50 Format: CSV Line Ending: Unix (LF) Include: ☒ header ☐ BOM

Buttons: GENERATE DATA, PREVIEW, SAVE AS..., DERIVE FROM EXAMPLE..., MORE

Figura 1 - Criação de Dados Fictícios

Criação de uma Base de Dados

Para a criação da base de dados, utilizei o site **Free MySQL Hosting**. Este serviço permite configurar uma base de dados MySQL de forma gratuita e rápida.

Database Details						
Database Host	Database Name	Database Username	Database Password	Database Size	Status	Delete
sql7.freemysqlhosting.net	sql7740525	sql7740525	Check your emails	0.02MB	Live	<input type="checkbox"/>
						Delete database

Figura 2 - Criação de uma Base de Dados

Transformações

Transformações referem-se ao conjunto de operações realizadas sobre os dados após a sua extração e antes do carregamento na base de dados ou sistema de destino. O objetivo das transformações é preparar os dados para que estejam em um formato adequado, consistente e utilizável para análise ou para outras operações.

Neste relatório, decidi incluir as transformações na descrição dos jobs, para que a interpretação dos leitores seja mais clara e completa. Ao detalhar cada transformação aplicada em cada job, espero proporcionar uma compreensão mais aprofundada do processo ETL e da importância de cada etapa na manipulação e preparação dos dados.

Jobs

1º Job – Carregamento, Transformação e Exportação de dados para uma Base de Dados

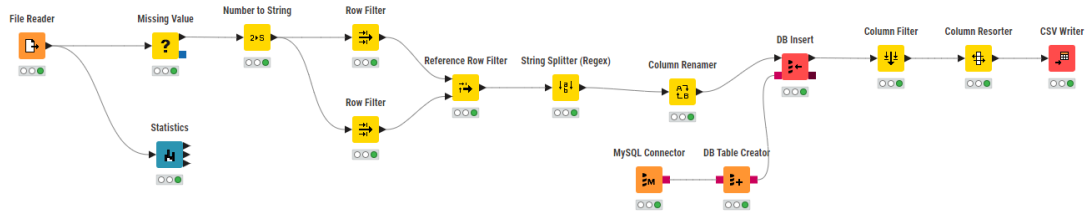


Figura 3 - 1º Job

Leitura de Ficheiro e Tratamento de campos vazios

O fluxo de trabalho começa com a leitura de um ficheiro de dados através do nó **File Reader**, que carrega o conjunto de dados inicial a ser processado. O primeiro passo na preparação dos dados é o tratamento de valores ausentes, utilizando o nó **Missing Value** para preencher os campos vazios na coluna "Age" com a média dos valores existentes, garantindo que a análise subsequente não seja comprometida (Configuração na figura 1).

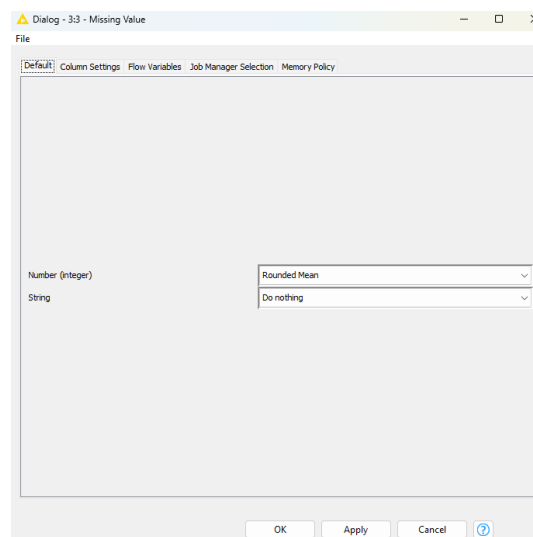


Figura 4 - Configuração de Missing Value

<input type="checkbox"/>	#	RowID	id Number (integ...	name String	age Number (integ...
<input type="checkbox"/>	13	Row12	13	Correy Gram...	17
<input type="checkbox"/>	14	Row13	14	Abbey Martin...	57
<input type="checkbox"/>	15	Row14	15	Merry Yitzhok	?
<input type="checkbox"/>	16	Row15	16	Kenna Ramsay	41
<input type="checkbox"/>	17	Row16	17	Seana Maudlin	40

Figura 5 - Dados Vazios

<input type="checkbox"/>	#	RowID	id Number (integ...	name String	age Number (
<input type="checkbox"/>	13	Row12	13	Correy Gram...	17
<input type="checkbox"/>	14	Row13	14	Abbey Martin...	57
<input type="checkbox"/>	15	Row14	15	14erry Yitzhok	30
<input type="checkbox"/>	16	Row15	16	Kenna Ramsay	41
<input type="checkbox"/>	17	Row16	17	Seana Maudlin	40

Figura 6 - Preenchimento dos Dados Vazios

Depois, aplico o nó **Number to String** para converter os valores das colunas "Id" e "Age" para formato string, facilitando a manipulação desses valores nas etapas seguintes.

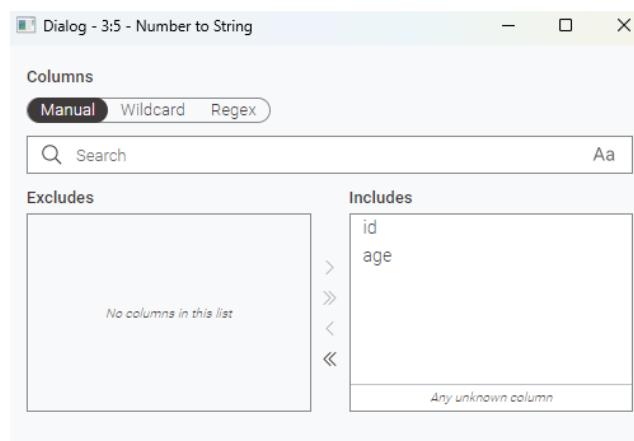


Figura 7 - Configuração de Number to String

<input type="checkbox"/>	#	RowID	id String	name String	age String
<input type="checkbox"/>	13	Row12	13	Correy Gram...	17
<input type="checkbox"/>	14	Row13	14	Abbey Martin...	57
<input type="checkbox"/>	15	Row14	15	Merry Yitzhok	30
<input type="checkbox"/>	16	Row15	16	Kenna Ramsay	41
<input type="checkbox"/>	17	Row16	17	Seana Maudlin	40

Figura 8 - Conversão de Inteiros para String

Filtragem e Segmentação de Dados

Na fase de filtragem e segmentação de dados, o **Row Filter** é utilizado duas vezes: primeiro, para filtrar os dados com base no gênero, selecionando apenas indivíduos do sexo feminino. Em seguida, aplico outra instância do **Row Filter** para escolher aqueles com idades entre 20 e 35 anos. Para isso, utilizei a expressão regular **([^]1[8-9]|2[0-9]|3[0-5])\$**), que permite identificar e selecionar idades válidas dentro desse intervalo.

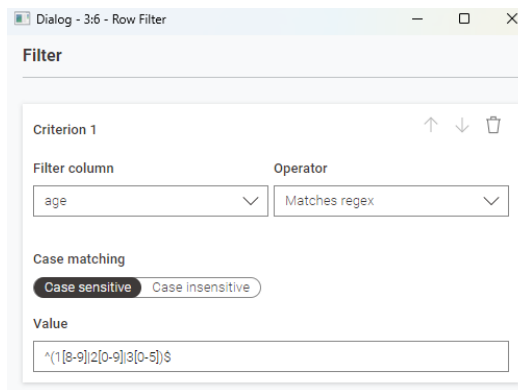


Figura 9 - Filtragem de Dados por Idade

Para filtrar por Gênero é utilizada a expressão regular **([^](?*i*)female\$**), para garantir que apenas os indivíduos sexo feminino serão contidos nos dados finais.

Dialog - 3:7 - Row Filter

Filter

Criterion 1

Filter column: gender

Operator: Matches regex

Case matching: **Case sensitive** Case insensitive

Value: ^(\?)female\$

Figura 10 - Filtragem por Género

Após essas duas etapas de filtragem, uso o **Reference Row Filter** para combinar os resultados e obter o conjunto final de dados que corresponde às mulheres com idades entre 20 e 35 anos, garantindo assim que os registos sejam relevantes para o estudo. Neste caso, a filtragem por género é utilizada para estabelecer a tabela filtrada por género como referência à tabela filtrada por idade.

Dialog - 3:8 - Reference Row Filter

Data column: gender

Reference column: gender

Include or exclude rows from the reference table: **Include** Exclude

Figura 11 - Filtragem por Referência

<input type="checkbox"/>	#	RowID	id String	name String	age String	email String	gender String
<input type="checkbox"/>	1	Row0	1	Evelyn Cattach	33	ecattach0@wi...	Female
<input type="checkbox"/>	2	Row2	3	Rayshell Rozea	32	rrozea2@mit...	Female
<input type="checkbox"/>	3	Row5	6	Minetta Talboy	25	mtalboy5@m...	Female
<input type="checkbox"/>	4	Row7	8	Reiko Benallack	26	rbenallack7@...	Female
<input type="checkbox"/>	5	Row9	10	Melesa Wollen	18	mwollen9@go...	Female

Figura 12 - Filtragem por Género e Idade

Transformação de Dados

De seguida, aplico o nó **String Splitter** com suporte a expressões regulares (regex) para dividir a coluna "Nome" em duas novas colunas, "First Name" e "Last Name", melhorando a estrutura dos dados. Para realizar essa divisão, utilizei a expressão regular **(^[A-Za-z-]+) +([A-Za-z-]+)\$**, que permite capturar os componentes do nome de forma adequada. A regex identifica o primeiro nome e o sobrenome, separando-os em colunas distintas.

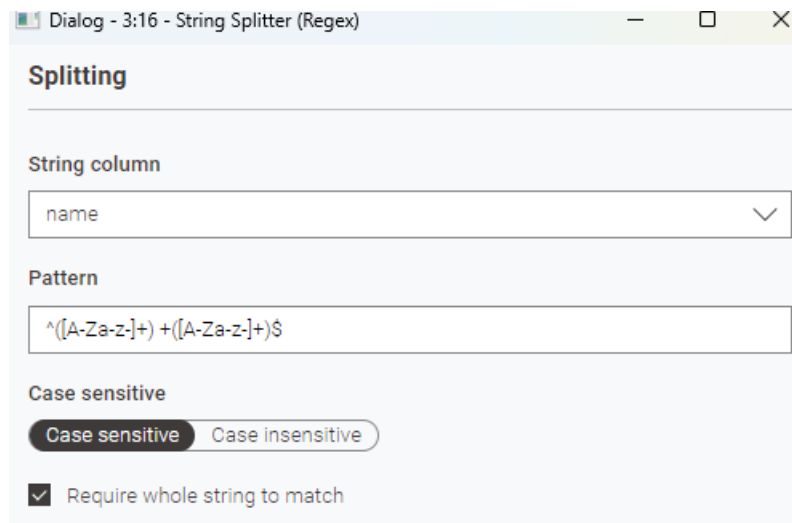


Figura 13 - Configuração de String Splitter

Inserção e Exportação de Dados

Para a inserção e exportação de dados, começando por estabelecer uma conexão com a base de dados MySQL através do nó **MySQL Connector**, que garante a correta execução das operações de inserção.

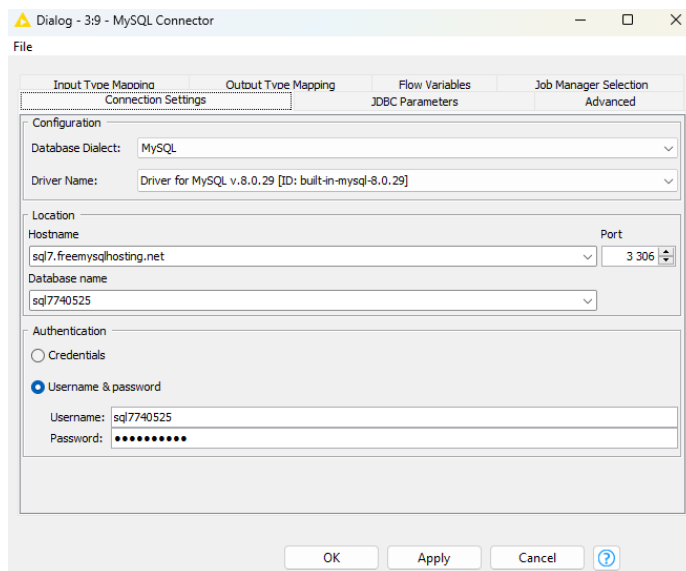


Figura 14 - Conexão à base de dados

Utilizo o **DB Table Creator** para criar a estrutura de tabelas necessária na base de dados, preparando-a para receber os dados transformados de forma organizada.

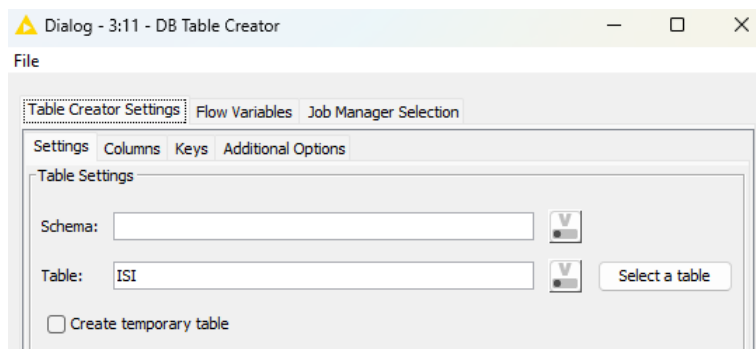


Figura 15 - Criação da Tabela "ISI"

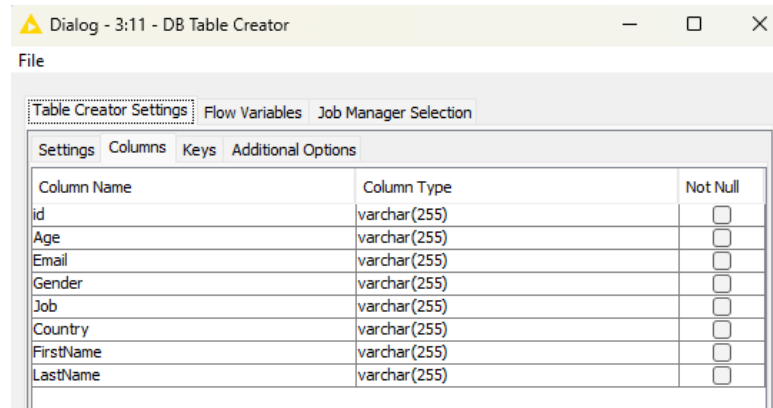


Figura 16 - Criação das tabelas Necessárias

Em seguida, o nó **DB Insert** é utilizado para inserir os dados processados diretamente na base de dados MySQL, assegurando que os dados sejam armazenados permanentemente para consultas ou análises futuras.

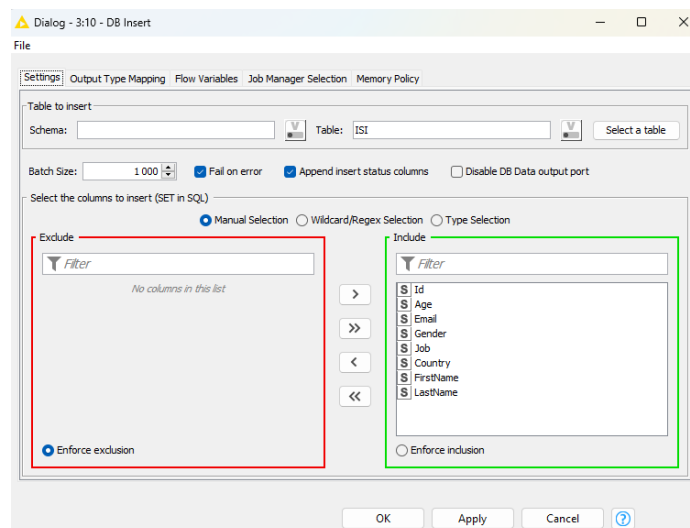


Figura 17 - Configuração de DB Insert

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

A mostrar registros de 0 - 24 (36 total. A consulta demorou 0.1512 segundos.)

SELECT * FROM 'ISI'

1 > >> | ☐ Mostrar tudo | Número de registros: 25 | Filtrar registros: Pesquisar esta tabela

id	Age	Email	Gender	Job	Country	FirstName	LastName
1	33	ecattach0@wikipedia.org	Female	Staff Accountant I	Sweden	Evelyn	Cattach
3	32	mrozea2@mit.edu	Female	Administrative Assistant I	Brazil	Rayshell	Rozea
6	25	mtalboy5@msu.edu	Female	Marketing Assistant	Finland	Minetta	Talboy
8	26	rbenallack7@taobao.com	Female	VP Quality Control	China	Reiko	Benallack
10	18	mwoollen9@google.it	Female	Developer II	Slovenia	Melesa	Wollen
20	22	dreiglarj@furl.net	Female	Recruiting Manager	Colombia	Dodi	Reiglar
23	31	jrosenn@discovery.com	Female	Biostatistician IV	Philippines	Joellen	Rosen
24	33	kedwinn@hugedomains.com	Female	Statistician III	Ukraine	Karalee	Edwin
37	27	hhubbins10@google.com	Female	Civil Engineer	Russia	Holly-anne	Hubbins
42	34	lcordeux15@blogger.com	Female	Design Engineer	Russia	Leisha	Cordeux
43	26	ocottel16@wordpress.com	Female	Mechanical Systems Engineer	Tanzania	Odelle	Cottel
48	28	rgreenrde1b@microsoft.com	Female	Marketing Manager	Indonesia	Ronni	Greenrde
1	33	ecattach0@wikipedia.org	Female	Staff Accountant I	Sweden	Evelyn	Cattach
3	32	mrozea2@mit.edu	Female	Administrative Assistant I	Brazil	Rayshell	Rozea
6	25	mtalboy5@msu.edu	Female	Marketing Assistant	Finland	Minetta	Talboy
8	26	rbenallack7@taobao.com	Female	VP Quality Control	China	Reiko	Benallack
10	18	mwoollen9@google.it	Female	Developer II	Slovenia	Melesa	Wollen
20	22	dreiglarj@furl.net	Female	Recruiting Manager	Colombia	Dodi	Reiglar
23	31	jrosenn@discovery.com	Female	Biostatistician IV	Philippines	Joellen	Rosen
24	33	kedwinn@hugedomains.com	Female	Statistician III	Ukraine	Karalee	Edwin
37	27	hhubbins10@google.com	Female	Civil Engineer	Russia	Holly-anne	Hubbins
42	34	lcordeux15@blogger.com	Female	Design Engineer	Russia	Leisha	Cordeux
43	26	ocottel16@wordpress.com	Female	Mechanical Systems Engineer	Tanzania	Odelle	Cottel
48	28	rgreenrde1b@microsoft.com	Female	Marketing Manager	Indonesia	Ronni	Greenrde
1	33	ecattach0@wikipedia.org	Female	Staff Accountant I	Sweden	Evelyn	Cattach

1 > >> | ☐ Mostrar tudo | Número de registros: 25 | Filtrar registros: Pesquisar esta tabela

Figura 18 - Base de Dados Preenchida

Filtragem e Reorganização de Colunas

Após a inserção dos dados, aplico o nó **Column Filter** para remover colunas desnecessárias, como "Insert Status" e "Error Status", que são geradas durante o processo de inserção, mas que não são relevantes para a análise.

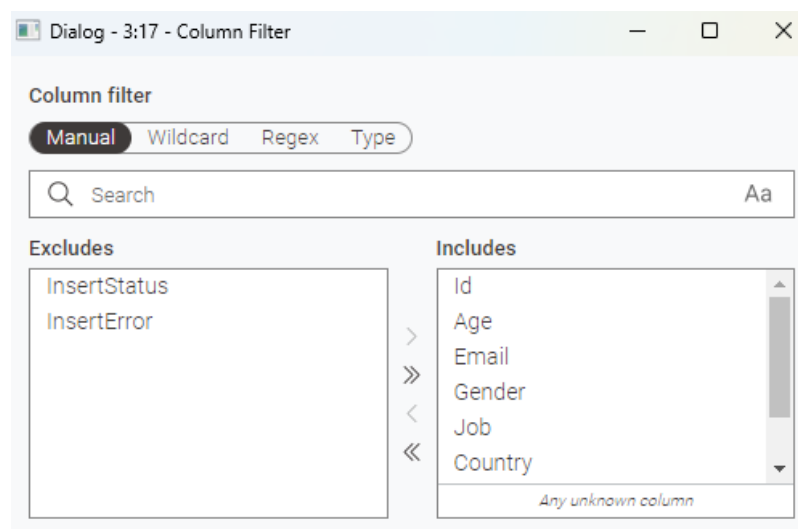


Figura 19 - Filtragem de Tabelas Desnecessárias

Em seguida, utilizo o Column Resorter para reorganizar as colunas numa ordem mais lógica e conveniente, facilitando a leitura e interpretação dos dados.

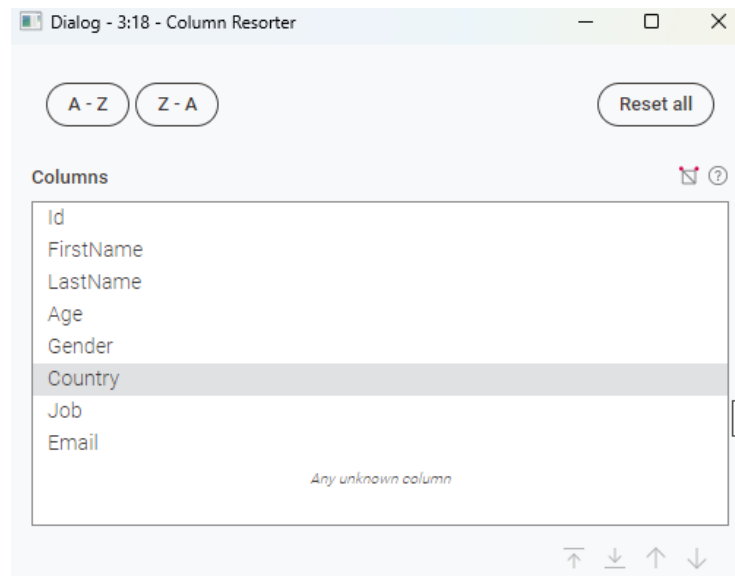


Figura 20 - Ordenação de Colunas

Exportação Final

Finalmente, o nó **CSV Writer** é utilizado para exportar os dados finais para um ficheiro CSV, especificando um caminho relativo. Isso garante que o processo de exportação possa ser facilmente executado noutro computador, independentemente da localização do ficheiro.

O formato CSV é ideal para facilitar a partilha e reutilização dos dados transformados, permitindo que sejam integrados com outras ferramentas e sistemas.

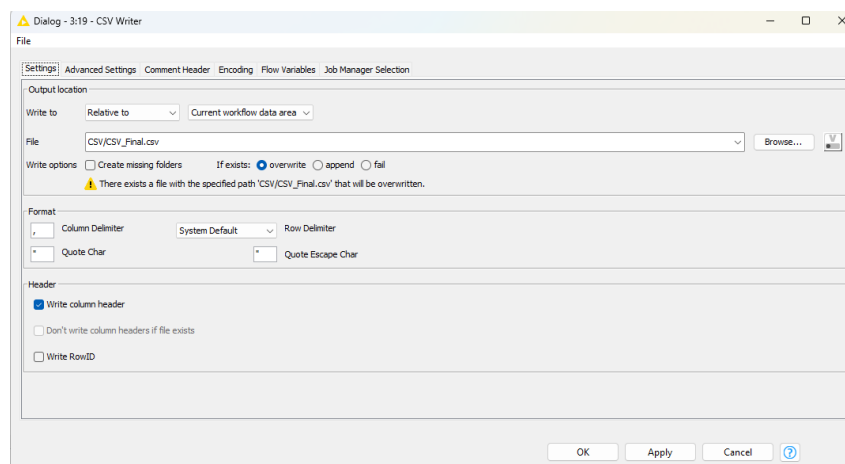


Figura 21 - Exportação do Ficheiro CSV

2º Job - Conversão de CSV para XML:



Figura 22 - Processo de Conversão de CSV para XML

Filtragem de Colunas:

O processo começa pela utilização do **Column Filter**, onde inicialmente são incluídas apenas duas colunas. Esta seleção reduz o volume de dados temporariamente, agilizando o processamento e evitando que o processo se torne demasiado demorado.

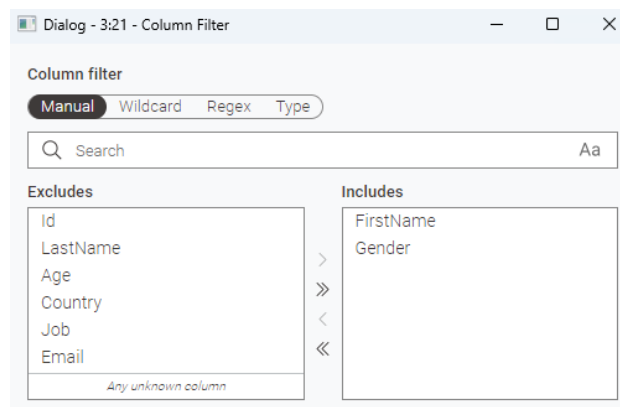


Figura 23 - Column Filter

<input type="checkbox"/>	#	RowID	FirstName String	<input type="checkbox"/>	Gender String
<input type="checkbox"/>	1	Row0	Evelyn	<input type="checkbox"/>	Female
<input type="checkbox"/>	2	Row1	Rayshell	<input type="checkbox"/>	Female
<input type="checkbox"/>	3	Row2	Minetta	<input type="checkbox"/>	Female
<input type="checkbox"/>	4	Row3	Reiko	<input type="checkbox"/>	Female
<input type="checkbox"/>	5	Row4	Melesa	<input type="checkbox"/>	Female

Figura 24 - Tabela Resultante de Column Filter

Conversão de Colunas Para XML

A seguir, utiliza-se o **Column to XML** para converter cada uma das colunas filtradas para o formato XML. Esta transformação permite que cada coluna seja convertida individualmente, criando uma estrutura XML básica que facilita a manipulação dos dados.

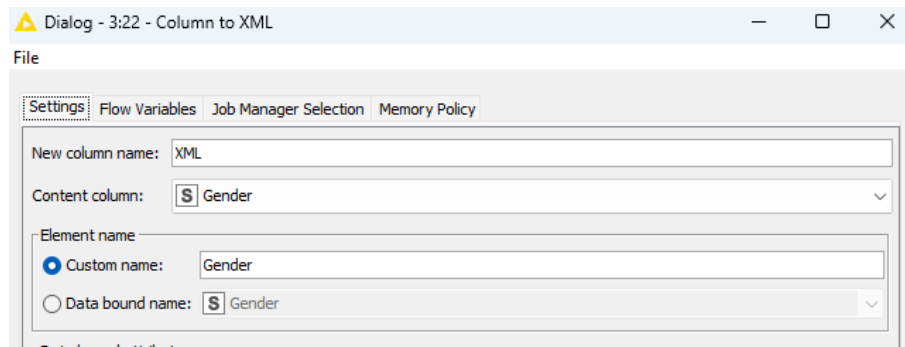


Figura 25 - Configuração de Column to XML Para Género

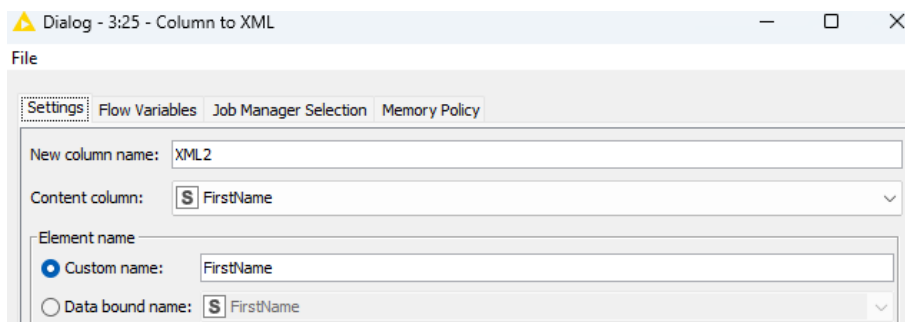


Figura 26 - Configuração de Column to XML Para First Name

	#	RowID	FirstName	XML
			String	
<input type="checkbox"/>	1	Row0	Evelyn	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	2	Row1	Rayshell	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	3	Row2	Minetta	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	4	Row3	Reiko	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	5	Row4	Melesa	<?xml version="1.0" encoding="UTF-8"?>

Figura 27 - Tabela Resultante da Conversão de Gender para XML

	#	RowID	XML	XML2
			String	
<input type="checkbox"/>	1	Row0	<?xml version="1.0" encoding="UTF-8"?>	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	2	Row1	<?xml version="1.0" encoding="UTF-8"?>	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	3	Row2	<?xml version="1.0" encoding="UTF-8"?>	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	4	Row3	<?xml version="1.0" encoding="UTF-8"?>	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	5	Row4	<?xml version="1.0" encoding="UTF-8"?>	<?xml version="1.0" encoding="UTF-8"?>

Figura 28 - Tabela Resultante da Conversão de First Name para XML

Combinação de Colunas

Depois da conversão de cada coluna, aplica-se o **XML Column Combiner** e o **XML Row Combiner**. Estes passos combinam as colunas e linhas em XML, unificando toda a informação e criando uma estrutura XML consolidada e bem formatada. Esta combinação das colunas e das linhas melhora significativamente a legibilidade e a organização dos dados, proporcionando um XML que segue uma estrutura lógica e alinhada com as exigências de formatação.

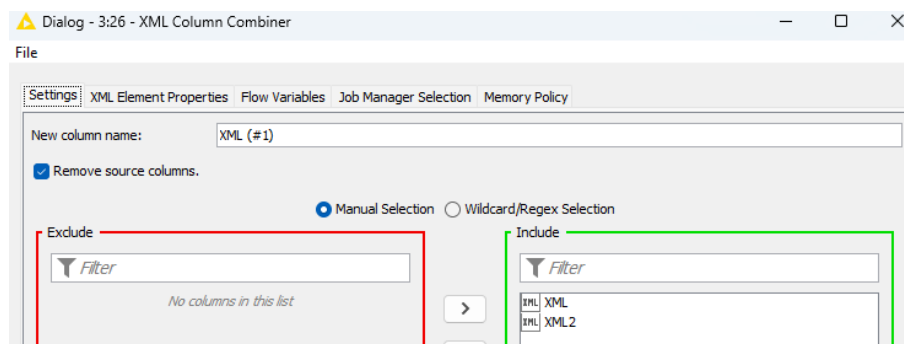


Figura 29 - Configuração de Column Combiner

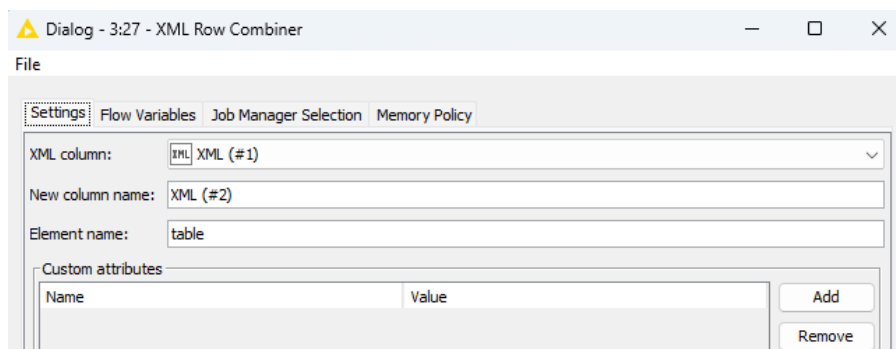


Figura 30 - Configuração de Row Combiner

<input type="checkbox"/>	#	RowID	XML (#1) XML
<input type="checkbox"/>	1	Row0	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	2	Row1	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	3	Row2	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	4	Row3	<?xml version="1.0" encoding="UTF-8"?>
<input type="checkbox"/>	5	Row4	<?xml version="1.0" encoding="UTF-8"?>

Figura 31 - Resultado de Column Combiner

<input type="checkbox"/>	#	R... ↓	XML (#2) XML
<input type="checkbox"/>	1	Row0	<?xml version="1.0" encoding="UTF-8"?>

Figura 32 - Resultado de Row Combiner

Escrita do Ficheiro XML

Por fim, utiliza-se o **XML Writer** para escrever o ficheiro XML final de forma permanente, guardando-o no sistema de ficheiros através de um caminho relativo. Esta última etapa permite que o ficheiro XML seja armazenado de forma acessível, pronto para ser utilizado em aplicações futuras ou para ser integrado com outros sistemas.

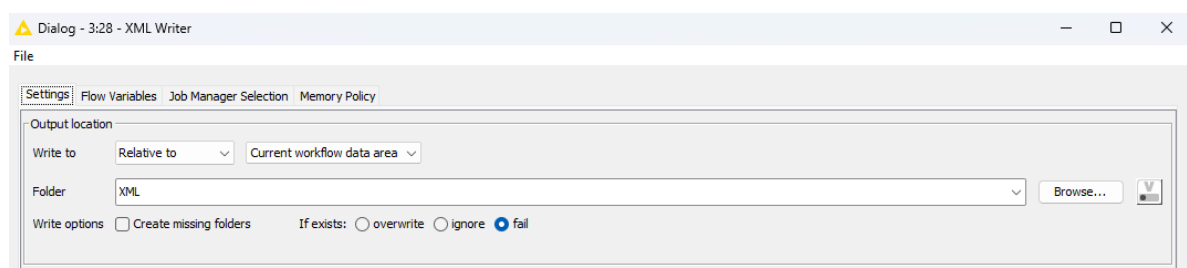


Figura 33 - Configuração do XML Witter

```
<?xml version="1.0" encoding="UTF-8"?>
<table>
  <row>
    <Gender>Female</Gender>
    <FirstName>Evelyn</FirstName>
  </row>
  <row>
    <Gender>Female</Gender>
    <FirstName>Rayshell</FirstName>
  </row>
  <row>
    <Gender>Female</Gender>
    <FirstName>Minetta</FirstName>
  </row>
</table>
```

Figura 34 - XML Resultante da Conversao de CSV

3º Job - Envio de Tabela Por Email



Figura 35 - Envio de Tabela Por Email

Leitura do Ficheiro CSV

O processo começa com a leitura de um ficheiro CSV utilizando o **CSV Reader**. Esta etapa importa os dados do ficheiro para uma tabela que pode ser manipulada no fluxo de trabalho.

Conversão de Tabela para HTML

Em seguida, utiliza-se o **Table to HTML String**, uma transformação que converte a tabela importada para o formato HTML. Essa conversão é essencial para que a tabela possa ser inserida de forma adequada no corpo do email, garantindo que a formatação e a estrutura dos dados fiquem visualmente organizadas e claras para o destinatário.

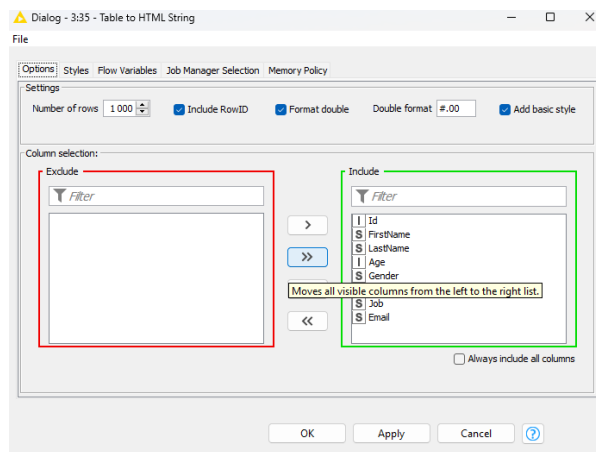


Figura 36 - Configuração de Table to HTML

Rows: 1 | Columns: 2

#	RowID	HTML String
1	HTML	<html>

Table Statistics Images List

Figura 37 - Resultado da Conversão para HTML

Envio do Email

Finalmente, configura-se o serviço de email com um componente de envio, onde os dados convertidos para HTML são incluídos diretamente no corpo do email. Com esta configuração, a tabela é enviada de forma estruturada e formatada, permitindo que o destinatário visualize as informações de maneira limpa e acessível, sem a necessidade de abrir um anexo adicional.

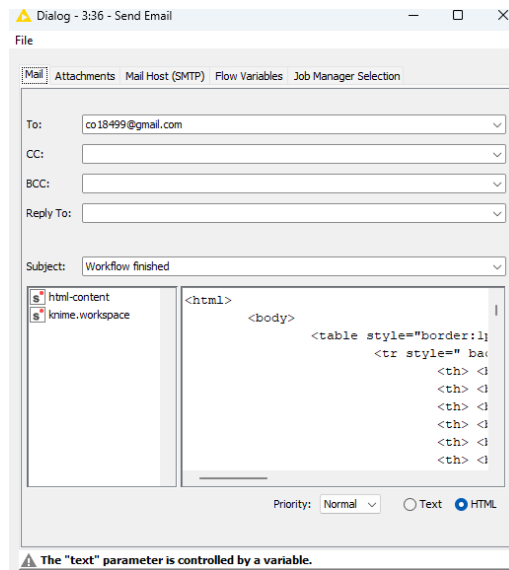


Figura 38 - Configuração do Remetente

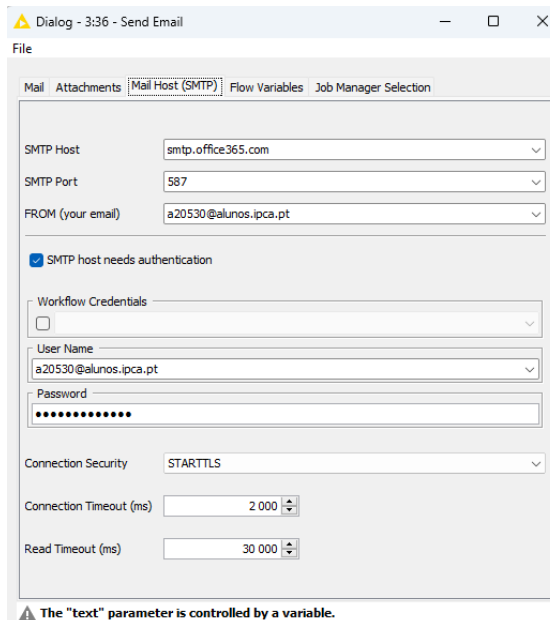


Figura 39 - Configuração do Cliente SMTP

Para integrar o conteúdo HTML no corpo do email, foi necessário criar uma **flow variable**. Esta variável permite armazenar o conteúdo gerado pelo **Table to HTML String** e utilizá-lo em etapas posteriores do job. A flow variable atua como um container temporário que armazena dados específicos ao longo do fluxo de trabalho, permitindo que sejam facilmente aplicados noutras operações.

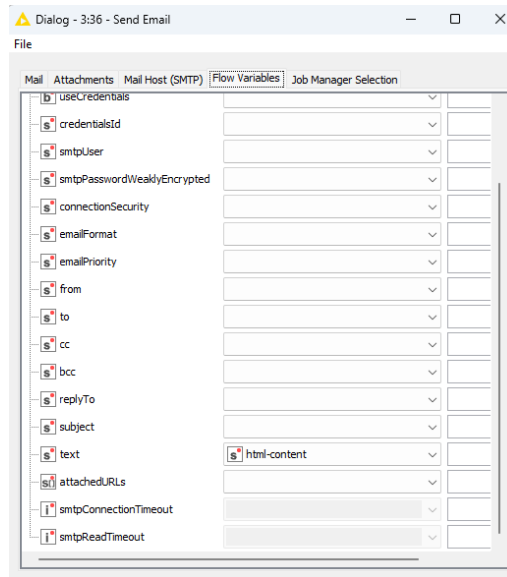


Figura 40 - Criação de uma Flow Variable

Workflow finished Caixa de entrada x

a a20530@alunos.ipca.pt

a a20530@alunos.ipca.pt
para mim ▾

RowID	Id	FirstName	LastName	Age	Gender	Country	Job	Email
Row0	1	Evelyn	Cattach	33	Female	Sweden	Staff Accountant I	ecattach0@wikipedia.org
Row1	3	Rayshell	Rozea	32	Female	Brazil	Administrative Assistant I	rrozea2@mit.edu
Row2	6	Minetta	Talboy	25	Female	Finland	Marketing Assistant	mtalboy5@msu.edu
Row3	8	Reiko	Benallack	26	Female	China	VP Quality Control	rbenallack7@taobao.com
Row4	10	Melesa	Wollen	18	Female	Slovenia	Developer II	mwollen9@google.it
Row5	20	Dodi	Reiglar	22	Female	Colombia	Recruiting Manager	dreiglarj@furl.net
Row6	23	Joellen	Rosen	31	Female	Philippines	Biostatistician IV	jrosenm@discovery.com
Row7	24	Karalee	Edwin	33	Female	Ukraine	Statistician III	kedwinn@hugedomains.com
Row8	37	Holly-anne	Hubbins	27	Female	Russia	Civil Engineer	hhubbins10@google.com
Row9	42	Leisha	Cordeux	34	Female	Russia	Design Engineer	lcordeux15@blogger.com
Row10	43	Odelle	Cottel	26	Female	Tanzania	Mechanical Systems Engineer	ocottel16@wordpress.com
Row11	48	Ronni	Greenrde	28	Female	Indonesia	Marketing Manager	rgreenrde1b@microsoft.com

Figura 41 - Email Recebido

4º Job – Get Request a uma API de Coordenadas

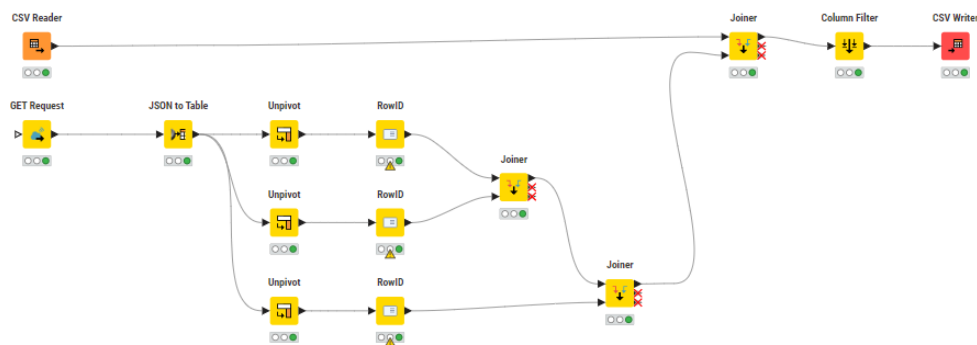


Figura 42 - Get Request a API

Extração de Dados

O processo começa com um **GET request** a uma API que retorna coordenadas geográficas de cidades portuguesas. Este passo permite obter os dados diretamente da API em formato JSON, contendo as informações necessárias sobre as localizações das cidades.

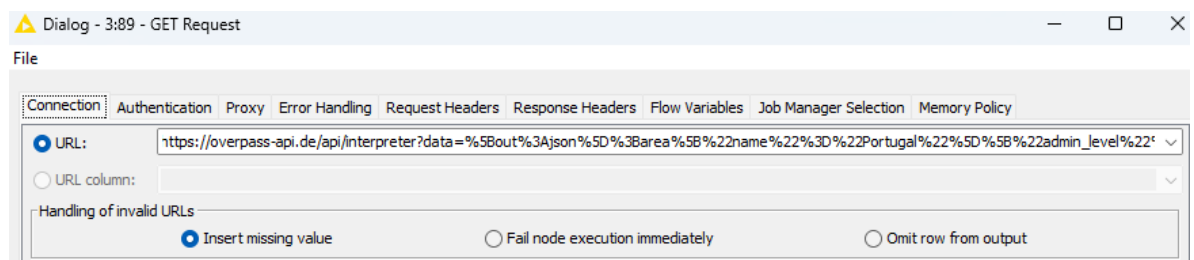


Figura 43 - URL do API no Get Request

Conversão de Json para Tabela

utiliza-se o **JSON to Table** para transformar o JSON retornado pela API numa tabela. Esta transformação facilita a manipulação dos dados, convertendo o formato JSON numa Tabela.

RowID	Status <small>Number (inte...)</small>	Content t... <small>String</small>	version <small>Number (dou...)</small>	generator <small>String</small>	timestam... <small>String</small>	timestam... <small>String</small>	copyright <small>String</small>	type <small>String</small>	id <small>Number (inte...)</small>	lat <small>Number (dou...)</small>
Row0	200	application/js...	0.6	Overpass API ...	2024-10-26T0...	2024-10-25T2...	The data incl...	node	24960091	40.657

Figura 44 - Tabela Resultante da Conversao de um Json

Conversão de colunas em linhas e identificação destas

O **Unpivot** é utilizado para converter colunas com o mesmo tipo de dados em linhas, armazenando-as numa única coluna. Esta técnica reorganiza os dados, permitindo que todos os valores de um determinado tipo sejam concentrados numa única coluna.

Value columns

Manual Wildcard Regex Type

Search Aa

Excludes

Status
Content type
version
generator
timestamp_osm_base
timestamp_areas_base
copyright

Includes

name
name (#1)
name (#2)
name (#3)
name (#4)
name (#5)

Any unknown column

Figura 45 - Unpivot das Colunas "Name"

#	RowID	RowIDs ↓ <small>String</small>	ColumnNames <small>String</small>	ColumnValues <small>String</small>
1	Row0	Row0	name	Viseu
2	Row1	Row0	name (#1)	Braga
3	Row2	Row0	name (#2)	Évora
4	Row3	Row0	name (#3)	Vila Real
5	Row4	Row0	name (#4)	Setúbal

Figura 46 - Tabela Resultante do Unpivot

O **RowId** é então utilizado para redefinir os IDs de cada coluna. Esta ação cria identificadores únicos e sequenciais para cada linha na tabela, o que facilita o uso desses IDs em operações de junção (joins) posteriores.

Replace RowIDs

☒ Replace RowIDs

Replacement mode

Generate new

Use column

ID column

RowIDs

☐ Remove selected ID column

If ID column contains missing values

Fail

Replace by "?"

If ID column contains duplicates

Fail

Append counter

Figura 47 - Configuração do RowId

<input type="checkbox"/>	#	RowID	RowIDs <small>String</small>	ColumnNames <small>String</small>	ColumnValues <small>String</small>
<input type="checkbox"/>	1	Row0	Row0	name	Viseu
<input type="checkbox"/>	2	Row...	Row0	name (#1)	Braga
<input type="checkbox"/>	3	Row...	Row0	name (#2)	Évora
<input type="checkbox"/>	4	Row...	Row0	name (#3)	Vila Real
<input type="checkbox"/>	5	Row...	Row0	name (#4)	Setúbal

Figura 48 - Tabela Resultante do RowId

Junção das Colunas Previamente Identificadas e filtragem das Colunas Desnecessárias

Posteriormente, o **Joiner** é aplicado para combinar as colunas que foram convertidas com o Unpivot, e, em passos adicionais, são utilizados outros **Joiner** para filtrar as cidades que foram importadas a partir de um ficheiro CSV, garantindo que o conjunto de dados final contenha apenas as cidades necessárias para o projeto.

Matching Criteria

Match

All of the following Any of the following

Top input ('left' table) Bottom input ('right' table)

city ColumnValues

+ Add matching criterion

Compare values in join columns by

Value and type

Figura 49 - Configuração do Joiner na Filtragem de dois Documentos

O **Column Filter** é então utilizado para seleccionar apenas as colunas essenciais, removendo dados desnecessários e deixando apenas a informação relevante para o próximo passo.

<input type="checkbox"/>	#	RowID	city String	<input type="checkbox"/>	ColumnValues (right) Number (double)	<input type="checkbox"/>	ColumnValues (right) (right) Number (double)
<input type="checkbox"/>	1	Row...	Viseu		40.657		-7.914
<input type="checkbox"/>	2	Row...	Braga		41.551		-8.428
<input type="checkbox"/>	3	Row...	Faro		37.016		-7.935
<input type="checkbox"/>	4	Row...	Barcelos		41.531		-8.619
<input type="checkbox"/>	5	Row...	Lisboa		38.708		-9.137

Figura 50 - Tabela Resultante da Filtragem de Colunas Desnecessárias

Exportação do Ficheiro CSV Resultante

Finalmente, o processo termina com a **exportação dos dados** para um ficheiro em formato CSV. Esta exportação é realizada diretamente para a memória do sistema, onde o ficheiro CSV resultante é criado e armazenado.

	A
1	city,"ColumnValues (right)","ColumnValues (right) (right)"
2	Viseu,40.6574713,-7.9138664
3	Braga,41.5510583,-8.4280045
4	Faro,37.0162727,-7.9351771
5	Barcelos,41.5314496,-8.6192306
6	Lisboa,38.7077507,-9.1365919
7	Aveiro,40.640496,-8.6537841
8	Viana do Castelo,41.694867,-8.831088

Figura 51 - Ficheiro CSV Exportado

Conclusão

Este trabalho prático permitiu consolidar o conhecimento adquirido na disciplina de Integração de Sistemas de Informação, focando na plataforma KNIME. Ao longo das atividades, foi possível gerar dados fictícios, transformar ficheiros de diferentes formatos e automatizar processos para facilitar a integração de sistemas de informação.

A aplicação de técnicas de conversão e transformação de dados ajudou a aprofundar a compreensão sobre a manipulação e filtragem de informações. Além disso, os jobs desenvolvidos demonstraram a importância de uma estrutura sequencial no processo ETL.

Referências Bibliográficas

1. KNIME Forum. (10/2024). Email a table. KNIME Hub.
<https://hub.knime.com/swebb/spaces/Public/LhasaNodes/EmailATable~MTTrnySntkM0qvEAN/current-state>
2. KNIME Forum. (10/2024). API GET request to JSON.
<https://forum.knime.com/t/api-get-request-to-json/67245>
3. KNIME Forum. (10/2024). How to obtain the output of GET request node in JSON format.
<https://forum.knime.com/t/how-to-obtain-the-output-of-get-request-node-in-json-format/9881/5>
4. KNIME Forum. (10/2024). Simple customisable XML generation from table. KNIME Hub.
<https://hub.knime.com/takbb/spaces/Public/Simple%20Customisable%20XML%20Generation%20from%20table~md1ubhfeFbJak9Uf/current-state>
5. KNIME Forum. (10/2024). Table to XML.
<https://forum.knime.com/t/table-to-xml/32705/3>
6. KNIME Forum. (10/2024). Table to JSON.
<https://forum.knime.com/t/table-to-json/38302/3>

Código QR do Vídeo de Demonstração



Figura 52 - Código QR do Vídeo de Demonstração