Relatório

Integração de Sistemas de Informação

Aluno/os: André Freitas nº21112
Eduardo Rebelo 21105

Professores: Luís Ferreira

Óscar Ribeiro

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, novembro, 2022

Resumo: Este trabalho da Disciplina de Integração de Sistemas de Informação (ISI) pretende-se focar a aplicação e experimentação de ferramentas em processos de ETL, com isso pretende-se que sejam desenvolvidos processos de ETL que envolvam scripts próprias ou que recorram a ferramentas disponíveis como o Pentaho Kettle, Microsoft SQL Server Integration Services (MSSIS), Knime, Talend open studio, ou outras.

Índice

1. Introdução 1

1.1. Principal problema a resolver 1

1.2. Objetivos do projeto 1

2. Estratégia utilizada 2

3. Primeira resolução (Kettle) 3

3.1. Resumo 3

3.2. MySQL Database 3

3.2.1 Tabelas 3

3.2.2. Conectividade com o Kettle 4

3.3. API 4

3.4. Job 5

3.5. Transformações 5

3.5.1. Transições 6

3.5.2. APIRates 6

3.5.3. CurrencyCountryNodes 7

3.5.4. Exchange Rate 8

3.5.5. Email 8

4. Segunda resolução (Knime) 9

4.1. Notas importantes 9

4.2. Explicação do trabalho 9

5. Pentaho Kettle vs Knime 23

6. Vídeo de demonstração 24

7. Conclusão 25

8. Bibliografia 26

Índice de figuras

Figura 1: Tabelas da base de dados 3
Figura 2: Conexão à base de dados 4
Figura 3: Job 5
Figura 4: Transições 6
Figura 5: Ligação à API 6
Figura 6: CurrencyCountryNodes 7
Figura 7: Conversão de moedas 8
Figura 8: Envio de email 8
Figura 9: Ficheiros CSV de leitura 9
Figura 10: Junção dos três ficheiros 10
Figura 11: Segundo workflow 11
Figura 12: Leitura da nova tabela e transformações 11
Figura 13: Tratamento das pequenas transações 12
Figura 14: Separação das duas décadas em quatro partes 13
Figura 15: Agrupamento de países distribuidores e recetores 13
Figura 16: Propriedades do agrupamento 13
Figura 17: Gráfico de barras 14
Figura 18: Conversão para JSON 14
Figura 19: Propriedades da distribuição 14
Figura 20: Resposta em JSON 15
Figura 21: Agrupamento e criação de ficheiro CSV 15
Figura 22: Propriedades de agrupamento 15
Figura 23: Ficheiro CSV 16
Figura 24: Última fase do segundo workflow 16
Figura 25: Propriedades dos agrupamentos 17
Figura 26: Ordenação de colunas 17
Figura 27: Propriedades do envio de email 18
Figura 28: Email enviado 18
Figura 29: Terceiro workflow 19
Figura 30: Início do workflow 19
Figura 31: Criação da tabela com o link da ligação à API 19
Figura 32: Método GET e XPath's 20
Figura 33: Propriedades do primeiro XPath 20
Figura 34: Propriedades do segundo XPath 20
Figura 35: Remoção de colunas 21
Figura 36: Ligação de base de dados e criação de tabela 21
Figura 37: Ligação à base de dados 21
Figura 38: Criação da tabela 22
Figura 39. Final do ciclo 22
Figura 40: Escrita dos dados na tabela 22
Figura 41: QR Code com vídeo de demonstração 24

1. Introdução

1.1. Principal problema a resolver

Com a atualidade da guerra da Ucrânia estamos habituados a ver notícias sobre as doações que os países aliados fazem para ajudar esta. Por isso achamos relevante ver as transações monetárias feitas entre países da América e da Europa que são os grandes aliados da ONU.

Para além disso, como falamos em tantos países, vamos automaticamente falar em diversas moedas diferentes, daí o outro problema deste projeto ser a conversão de todas as moedas estrangeiras para o Euro.

1.2. Objetivos do projeto

Existem dois principais objetivos:

- Manipulação de dados e transformações envolvendo as transações realizadas entre 2000 e 2020.
- Conversão em tempo real de todas as moedas registadas no banco central Europeu para a moeda Euro.

2. Estratégia utilizada

Como somos dois elementos num grupo tivemos de usar plataformas diferentes para a realização deste projeto daí normal as diferenças entre um e outro, algo que vai ser abordado mais à frente no relatório.
Com isto, o André teve de usar a plataforma “Knime” e o Eduardo o “Pentaho Kettle”.

Em ambas as plataformas foram abordados vários processos e operações, desde as coisas mais básicas como as transformações, os “Jobs”, agrupamentos, “lookups” e operações sobre valores, até tarefas mais complexas como o registo de informação numa base de dados, o envio de emails e a ligação a uma API de forma a recolher informação em tempo real.

É importante realçar que devido à diferença de plataforma, a maneira como se resolveu um desafio irá ser diferente, logo o uso de um determinado processo ou operação numa plataforma não obriga a que a outra faça o mesmo.

3. Primeira resolução (Kettle)

3.1. Resumo

O trabalho em “Kettle” resume-se em 5 transformações em 1 job, sendo neste último que se encontram todas as transformações bem como o processo de enviar email.

3.2. MySQL Database

O primeiro passo a ser tomado foi onde guardar os dados que recebemos e desta forma optou-se pela criação de uma base de dados em “MySql” com o nome de "transitionsdb".

3.2.1 Tabelas

A base de dados conta então com as 4 tabelas presentes na figura ao lado.
Sendo a “countryandcurrency” responsável por armazenar os códigos do país bem como os códigos das moedas desses países juntamente com o valor de cada moeda em relação ao euro.
A “currencyvalues” é uma tabela de apoio onde extraímos diretamente de uma API o código e o valor da moeda.
A tabela “transitions” é onde se encontra as transações iniciais, já a tabela “transitionseuros” é a tabela onde ficam guardadas todas as transações que foram possíveis converter para euro.

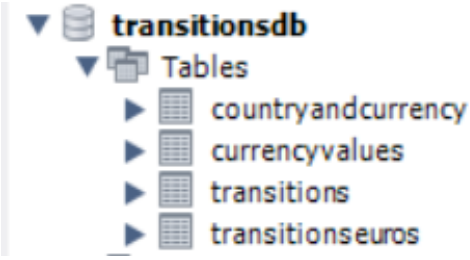


Figura 1: Tabelas da base de dados

3.2.2. Conectividade com o Kettle

A conectividade com o “Kettle” foi feita com os dados que são apresentados na imagem abaixo.

É de notar que esta conectividade foi necessária sempre que se usava um comando que interagia com a base de dados.

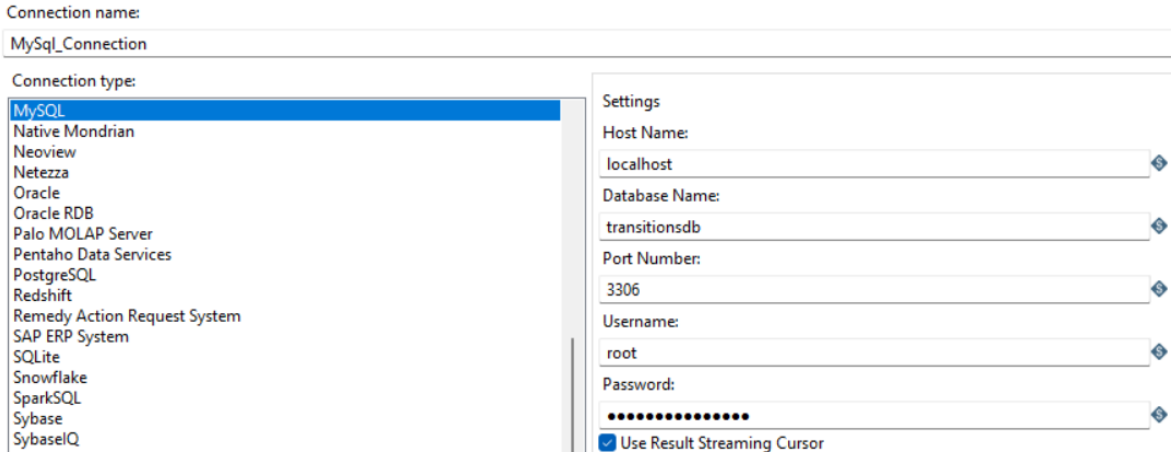


Figura 2: Conexão à base de dados

3.3. API

Por incentivo do professor tentamos estabelecer uma ligação com uma API que tivesse os valores das moedas ao longo dos anos, após várias tentativas sem sucesso de conectar a uma API com esses dados, decidi improvisar e criar uma API simples através do uso de um ficheiro XML (Pasta FILES, API.xls) e de uma ferramenta web que gera APIs.

3.4. Job

Assim como foi referido acima, foi utilizado unicamente um job, onde estão presentes todas as transformações e ainda o processo em que é enviado o email, como é possível verificar na imagem abaixo.

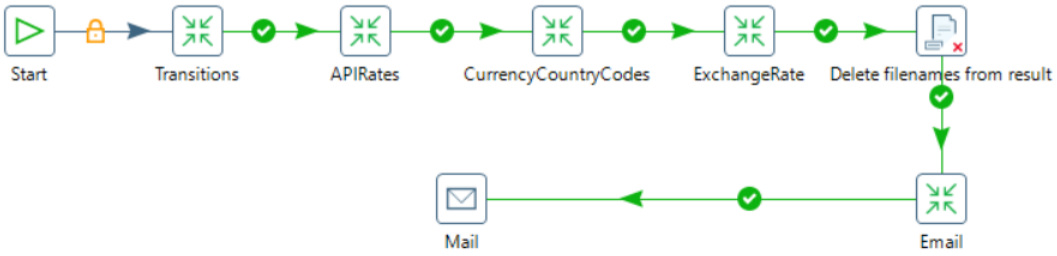


Figura 3: Job

3.5. Transformações

Nesta secção serão apresentadas todas as transformações utilizadas bem como uma breve descrição de cada.

3.5.1. Transições

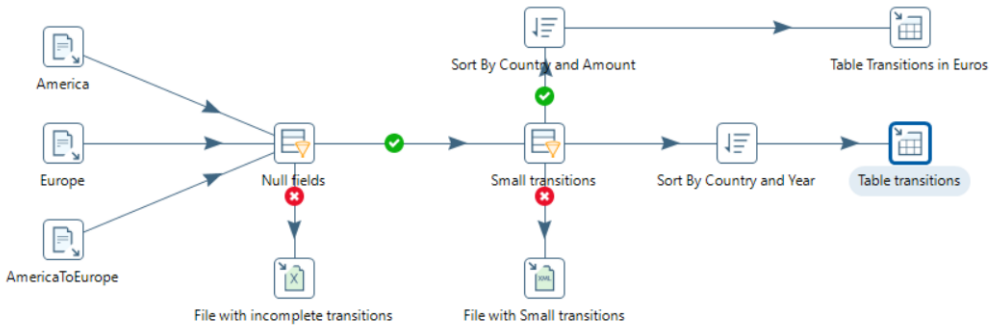


Figura 4: Transições

É nesta transformação que iremos receber três ficheiros CSV onde se encontram os dados de todas as transações (entre países europeus, entre países americanos e ainda entre os dois continentes).
Após receber os ficheiros iremos então filtrar os ficheiros e a partir dos dados excluídos criamos dois ficheiros (transações com dados incompletos e transações de montante baixa).
Os dados que forem aceites serão então ordenados e de seguida colocados nas tabelas de base de dados.

3.5.2. APIRates

Figura 5: Ligação à API

Como podemos ver na imagem acima, é nesta transformação que é feita a ligação à API, e que uma vez feita essa ligação, os dados são filtrados e ordenados para de seguida serem guardados numa tabela da base de dados.

3.5.3. CurrencyCountryNodes

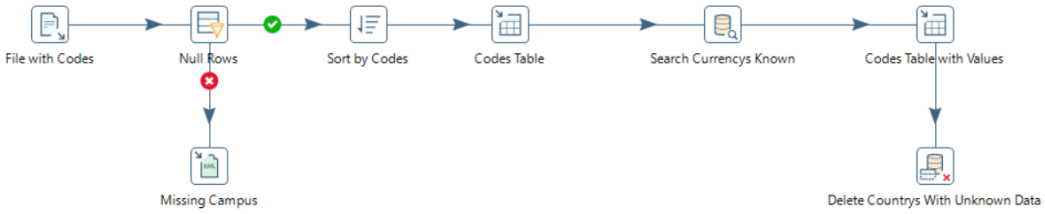


Figura 6: CurrencyCountryNodes

Nesta transformação recebemos os dados de um ficheiro onde estão contidos os códigos dos países juntamente com os códigos da moeda de cada país, verificámos campos vazios, ordenamos por ordem alfabética e de seguida armazenamos na base de dados.

Após os dados estarem guardados comparamos com os dados que obtivemos da API e adicionamos o valor de cada moeda para que de seguida os países que não temos a moeda sejam removidos.

3.5.4. Exchange Rate

Figura 7: Conversão de moedas

Esta transformação é possivelmente a mais importante de todas e também a que irá demorar mais no processo todo, uma vez que é nela que iremos calcular o valor das transações guardadas para euro e que, após isso, iremos consultar quais os valores ficaram vazios para que desta forma essas transações sejam removidas evitando uma sobrecarga de dados.

3.5.5. Email

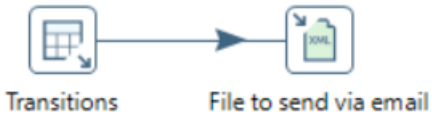


Figura 8: Envio de email

Nesta simples transação iremos só criar o ficheiro que irá ser enviado por e-mail.

4. Segunda resolução (Knime)

4.1. Notas importantes

Ao contrário do que acontece no “Pentaho Kettle”, não existe uma distinção entre transformações e jobs, todos os procedimentos são feitos num “workflow”, claro que podemos criar vários “workflows” de modo a organizar o projeto, o que por sinal acontece no nosso caso.

Para além disto fomos forçados a fazer um procedimento exclusivo ao “Knime” devido a uma limitação que este possui comparado com o “Kettle”. Este procedimento resultou do facto do “Knime” não conseguir ler mais do que um ficheiro ao mesmo tempo e imediatamente aplicar uma transformação nos três ao mesmo tempo.

4.2. Explicação do trabalho

Row ID	S CountryS	S CountryR	D Amount	I Year
Row0	LVA	LTU	4,602.58	2000
Row1	GRC	SVK	259.98	2012
Row2	ITA	BGR	7,098.32	2006
Row3	SVN	SWE	3,282.82	2005
Row4	SWE	MLT	7,096.78	2019
Row5	EST	DNK	3,985.83	2013
Row6	FRA	HUN	5,068.82	2012
Row7	DEU	DEU	235.53	2018
Row8	IRL	SVK	9,483.94	2006
Row9	FRA	BGR	36.81	2004
Row10	LTU	EST	9,545.09	2010
Row11	AUT	NLD	6,413.67	2018
Row12	FRA	ESP	8,071.13	2000
Row13	BEL	BEL	4,574.82	2017
Row14	CZE	NLD	7,026.2	2004
Row15	DEU	PRT	5,280.09	2017
Row16	HRV	CZE	1,511.89	2012
Row17	MLT	SVK	3,720.8	2004
Row18	SWE	POL	8,697.72	2019
Row19	ROU	DEU	5,595.99	2002
Row20	IRL	LVA	5,082.5	2011

Figura 9: Ficheiros CSV de leitura

Estes são os ficheiros .csv na qual irá ser recolhida a informação acerca das transações monetárias que iremos usar para fazer as nossas transformações.

Este diagrama representa exatamente o procedimento referido em cima sobre o facto de a plataforma não conseguir transformar mais de um ficheiro ao mesmo tempo.

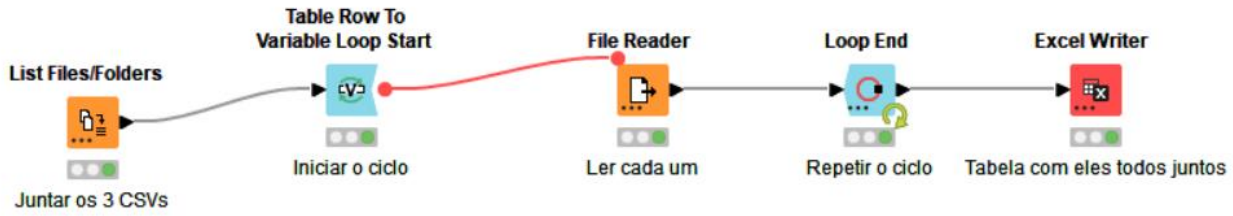


Figura 10: Junção dos três ficheiros

Primeiramente temos de juntar os três ficheiros num só e é exatamente isso o que o primeiro node faz. Este vai a uma pasta e recolhe todos os ficheiros dentro desta para poder listá-los todos numa só lista.

De seguida vamos começar um ciclo que vai permitir fazer a leitura de todos os ficheiros, dado que com este ciclo, o programa vai ler cada um dos ficheiros até eles estiverem todos lidos.

Depois de acabar este ciclo podemos finalmente registar toda a informação num só ficheiro que neste caso será uma tabela em XLS.

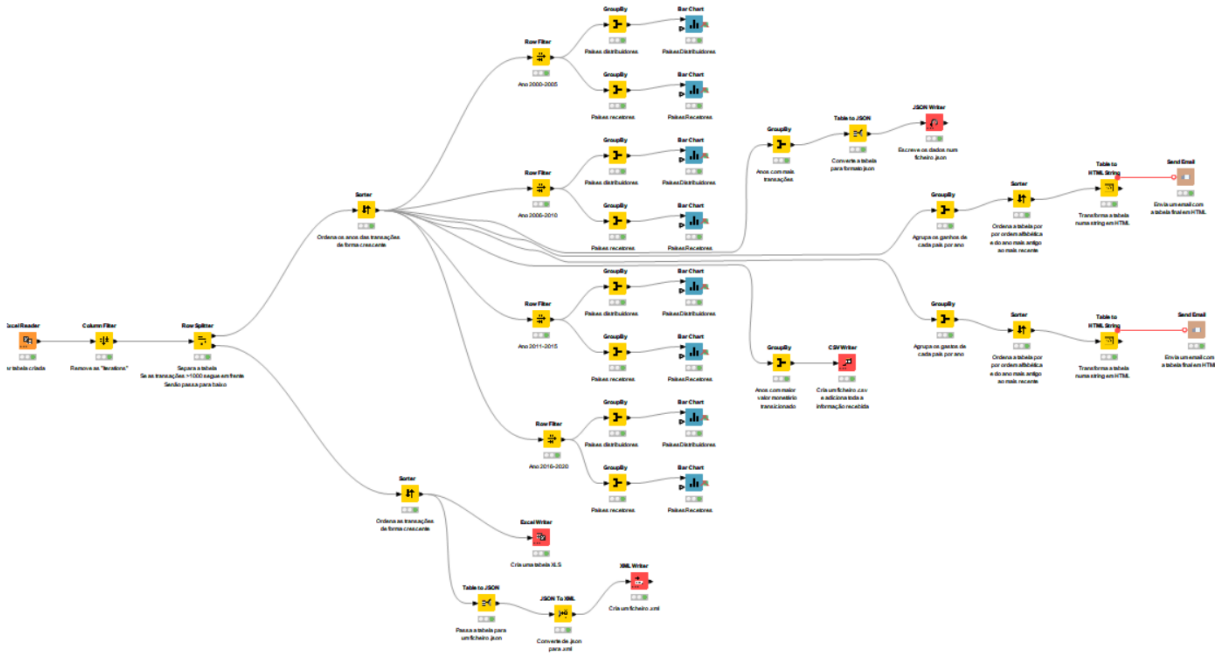


Figura 11: Segundo workflow

Este é o “workflow” encarregue por todas as transformações e os jobs envolvendo as transações monetárias das últimas duas décadas.

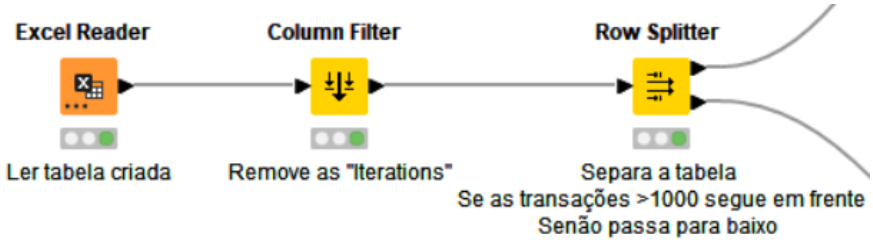


Figura 12: Leitura da nova tabela e transformações

Em primeiro lugar vamos ler a tabela criada no “workflow” anterior que guarda a informação dos três ficheiros CSV.

De seguida vamos usar um filtro de colunas para poder remover uma coluna da tabela que não tem relevância para o projeto.

Depois, vamos separar a tabela em duas partes. Se houver transações abaixo de mil, estas serão largadas para uma tabela própria. As outras transações permanecerão na tabela atual.

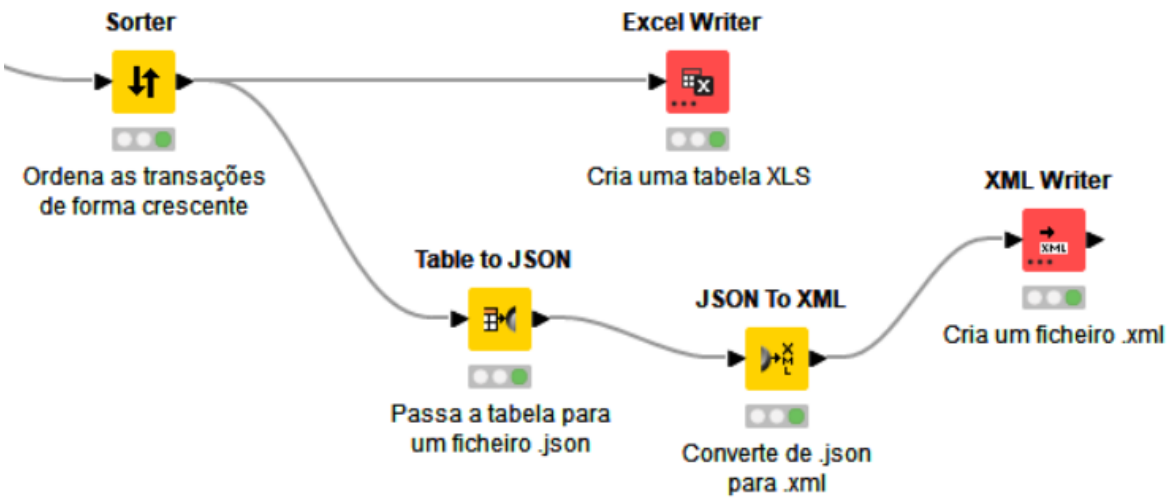
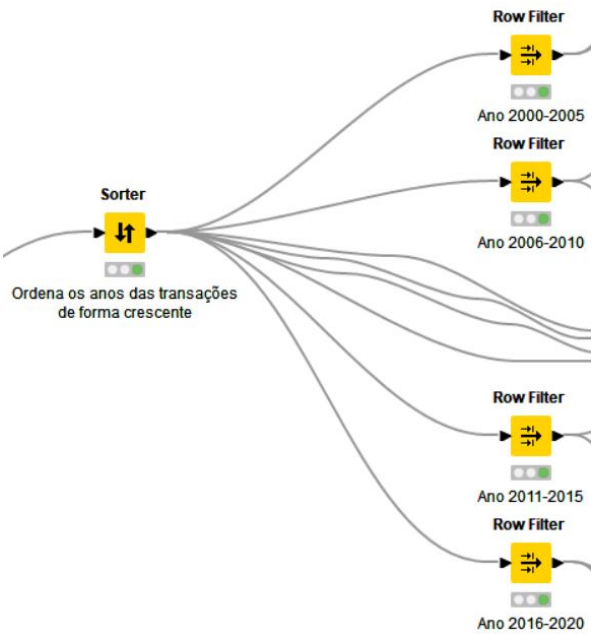


Figura 13: Tratamento das pequenas transações

A tabela criada com as pequenas transações vai ser ordenada das transações mais baixas às mais altas.



Para não deixar esta tabela abandonada vamos convertê-la para um ficheiro XLS e um ficheiro XML. Mas para passar a XML é necessário usar os nodes que transformam a tabela em código JSON e que passam o código JSON para um ficheiro XML. A tabela principal numa primeira estância vai ser ordenada consoante a ordem crescente dos anos em que as transações ocorreram. Depois disto vamos dividir as duas décadas em 4 partes de modo a facilitar a manipulação dos dados, bem como reduzir o tamanho da tabela.

Figura 14: Separação das duas décadas em quatro partes

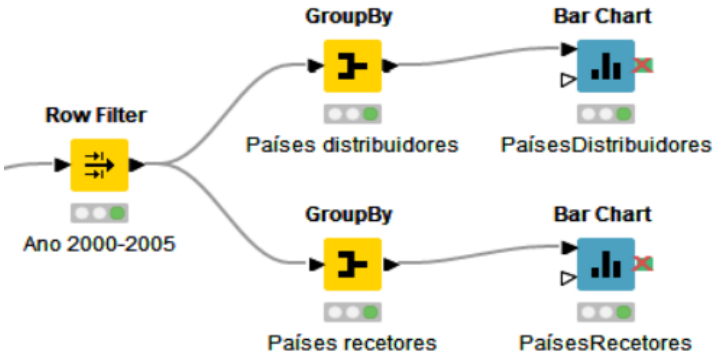


Figura 15: Agrupamento de países distribuidores e recetores

Pegando em cada parte vamos distribuí-la pelos países que enviam o dinheiro e pelos países que recebem o dinheiro. Para fazer isto usamos um agrupador que pega na coluna de cada tipo de país e na soma da coluna do montante relativo a cada país, como podemos verificar na imagem abaixo:

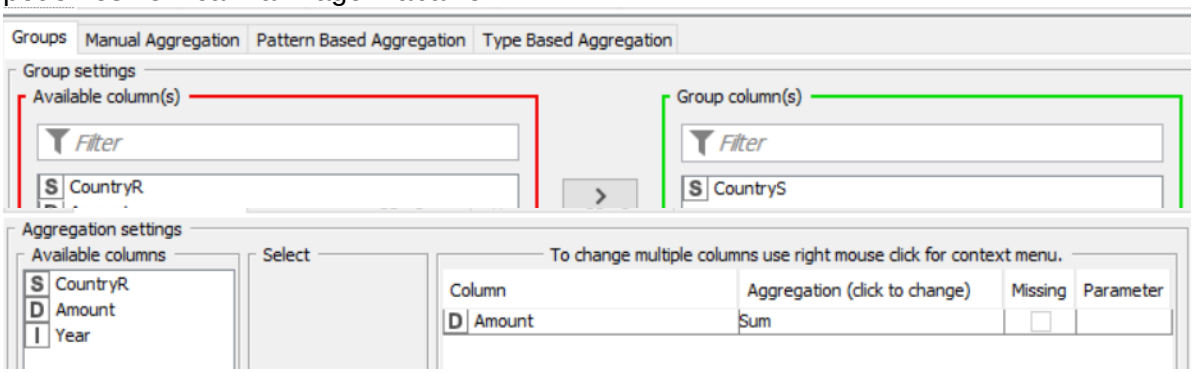


Figura 16: Propriedades do agrupamento

Para finalizar é criado um gráfico de barras que representa a informação organizada anteriormente. Isto facilita a visualização da distribuição dos dados.

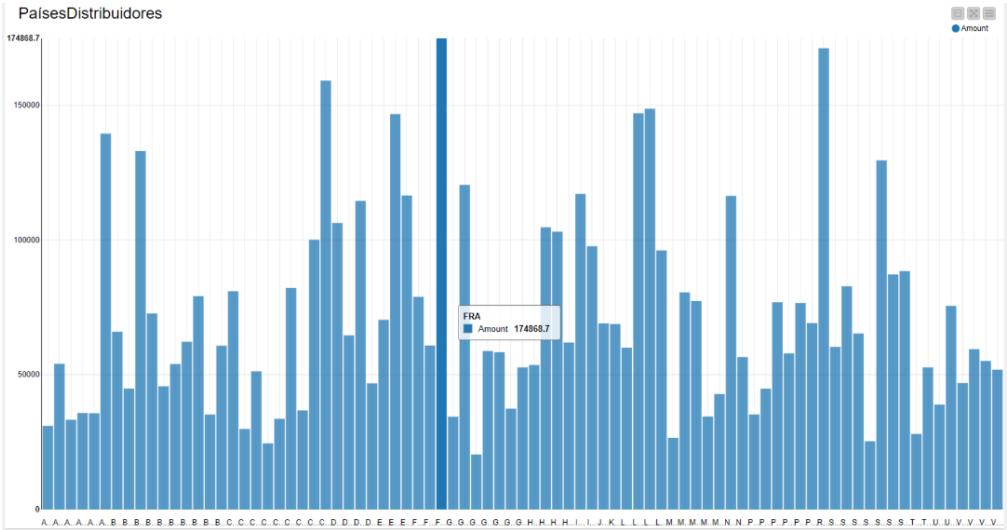


Figura 17: Gráfico de barras

Dado que já mostramos a distribuição das transações por cada país, é altura de revelar esta distribuição perante cada ano e é isso mesmo que fazemos de seguida.

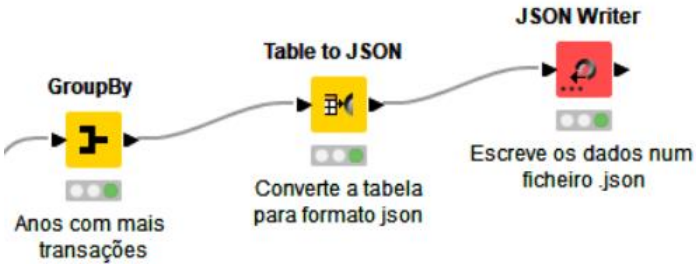


Figura 18: Conversão para JSON

Primeiro distribuímos pela quantidade de transações feitas em cada ano. Esta distribuição é feita pegando em todos os anos e atribuindo o número de transações ocorridas naquele ano, como podemos ver em baixo:

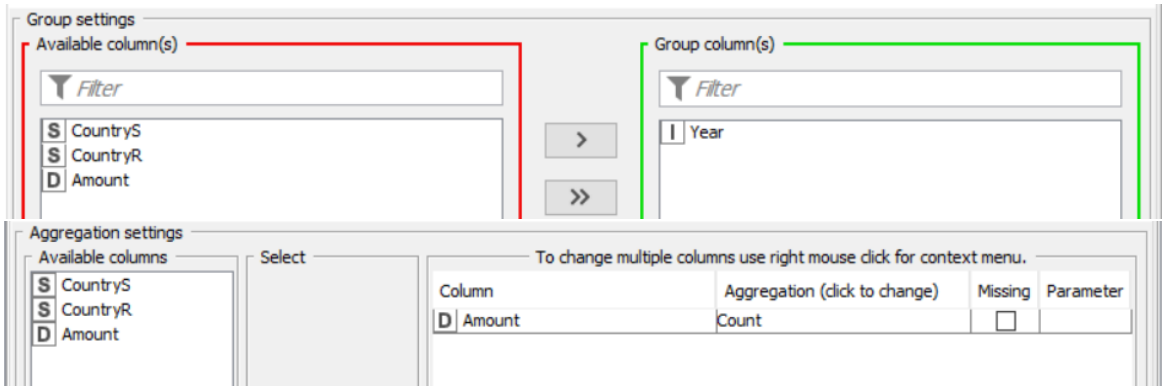


Figura 19: Propriedades da distribuição

Dado isto transcrevemos a tabela em código JSON, para poder depois criar um ficheiro JSON que vai guardar o código criado no node anterior.

Row ID	JSON	Output Location	Status
Row1	[{ "Year": 2000, "Amount": 181 }, { "Year": 2001, "Amount": 178 }, { "Year": 2002, "Amount": 190 },	../outputs/FileYearsTransactions_0.j...	created

Figura 20: Resposta em JSON

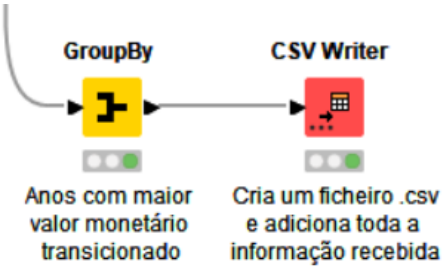


Figura 21: Agrupamento e criação de ficheiro CSV

Em contraste disto, é feito outro agrupamento, mas que neste caso vai calcular a soma do valor monetário de todas as transações num determinado ano.

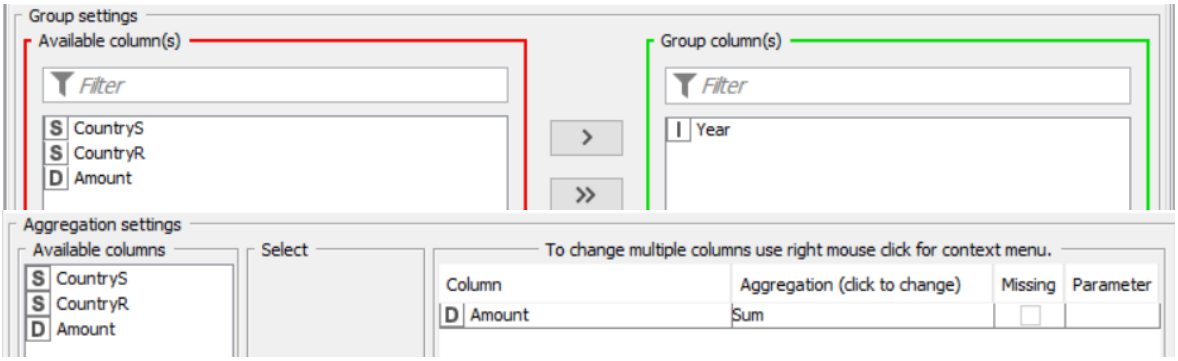


Figura 22: Propriedades de agrupamento

```
"Year","Amount"
2000,1010367.0800000001
2001,992112.2400000003
2002,1089619.8999999992
2003,966608.2099999997
2004,968985.1100000002
2005,957717.2200000006
2006,871441.9899999995
2007,854848.9300000002
2008,884742.2400000003
2009,794766.8000000003
2010,756834.4700000001
2011,930408.4100000004
2012,917650.1899999994
2013,798265.9000000003
2014,861048.8699999996
2015,949757.4800000004
2016,957265.3999999997
2017,1047147.8200000002
2018,974776.7400000001
2019,1029593.269999999
2020,1017474.6000000001
```

Figura 23: Ficheiro CSV

Para acabar esta sequência vamos criar um ficheiro CSV que vai conter todos os anos e o seu montante respetivo de transações.

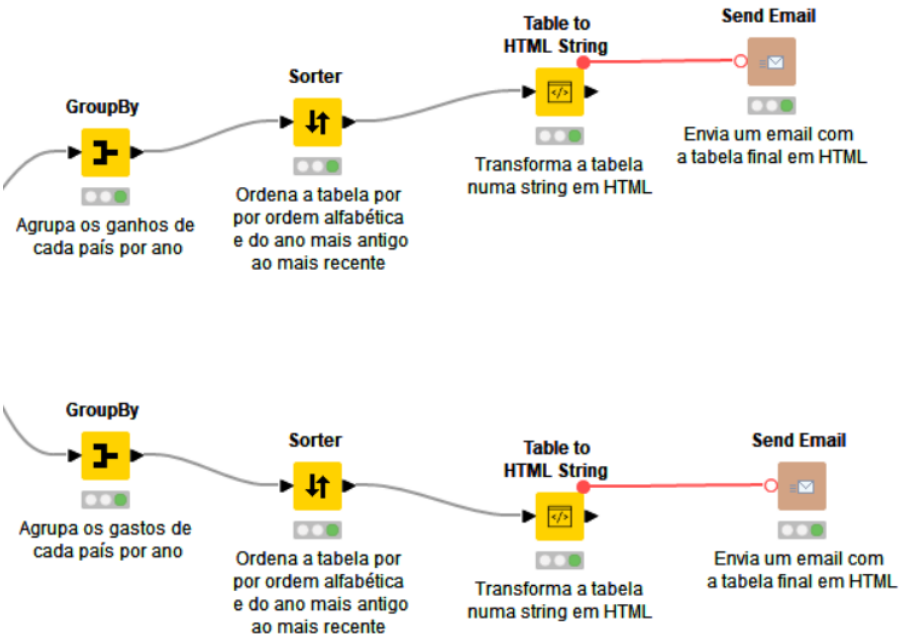


Figura 24: Última fase do segundo workflow

Para finalizar este “workflow” vamos criar duas tabelas finais. Sendo que uma delas está agrupada pelos ganhos de cada país em cada ano e a outra pelos gastos de cada país em cada ano.

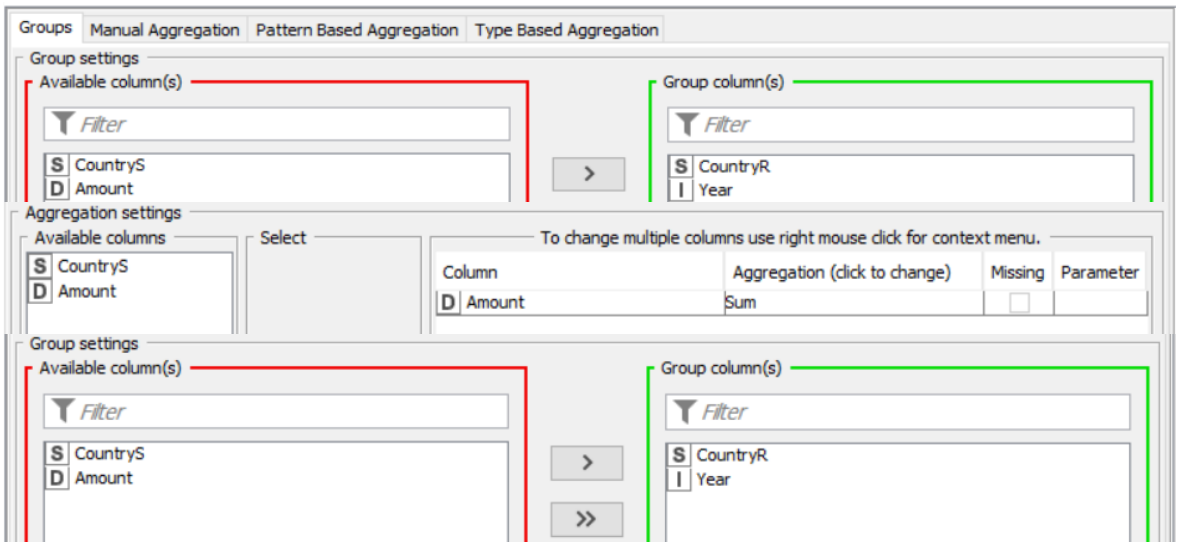
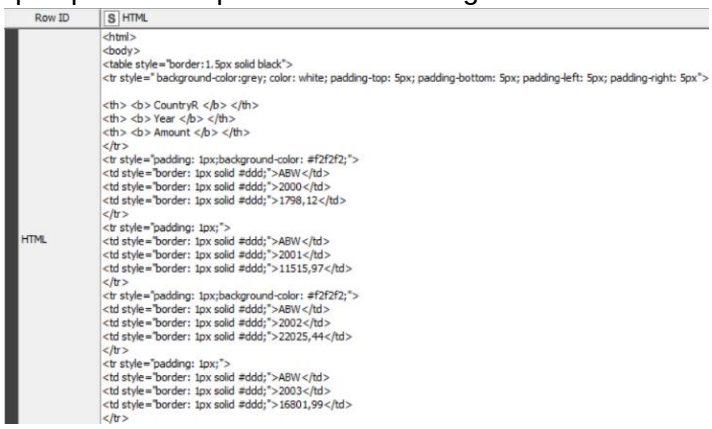


Figura 25: Propriedades dos agrupamentos

De seguida vamos ordenar, nas duas tabelas, a coluna dos países por ordem alfabética e a coluna do ano de forma crescente de forma a organizar a nossa informação.

Figura 26: Ordenação de colunas

O próximo node é muito especial pois este não se encontra no leque de nodes providenciados pelo “Knime”. É um node criado pela comunidade que permite passar qualquer tabela para linhas de código HTML.



ABW	2016	11790,35
ABW	2017	10091,13
ABW	2018	13492,68
ABW	2020	20048,91
AIA	2000	21957,78
AIA	2001	5171,61
AIA	2003	13969,27
AIA	2004	22865,41
AIA	2005	17177,21
AIA	2006	2895,84

Estas linhas de código vão gerar uma simples tabela com a informação final. Com isto somos capazes de mandar um mail que contenha esta tabela através da variável criada na tabela do código HTML.

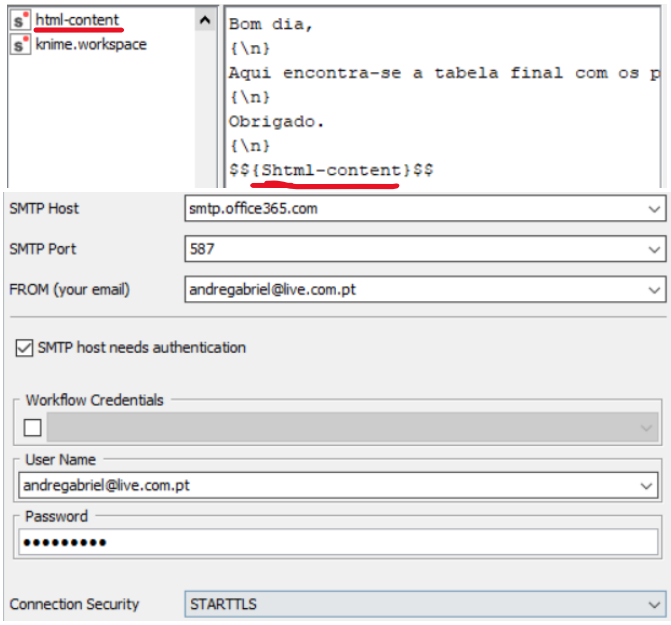
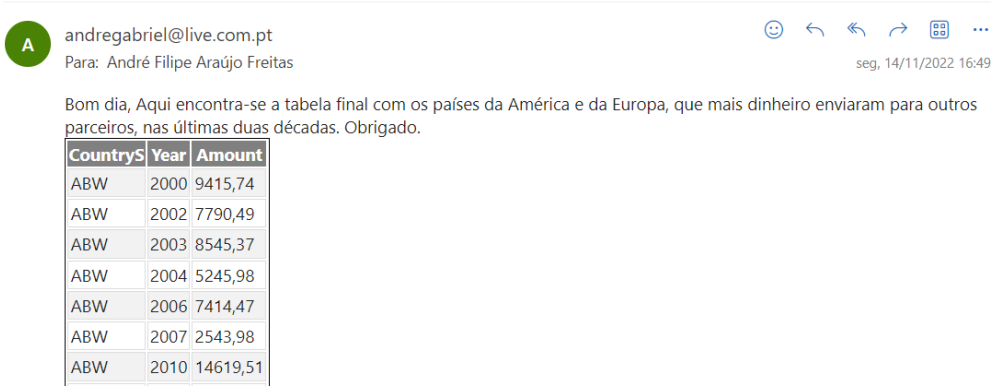


Figura 27: Propriedades do envio de email



CountryS	Year	Amount
ABW	2000	9415,74
ABW	2002	7790,49
ABW	2003	8545,37
ABW	2004	5245,98
ABW	2006	7414,47
ABW	2007	2543,98
ABW	2010	14619,51

Figura 28: Email enviado

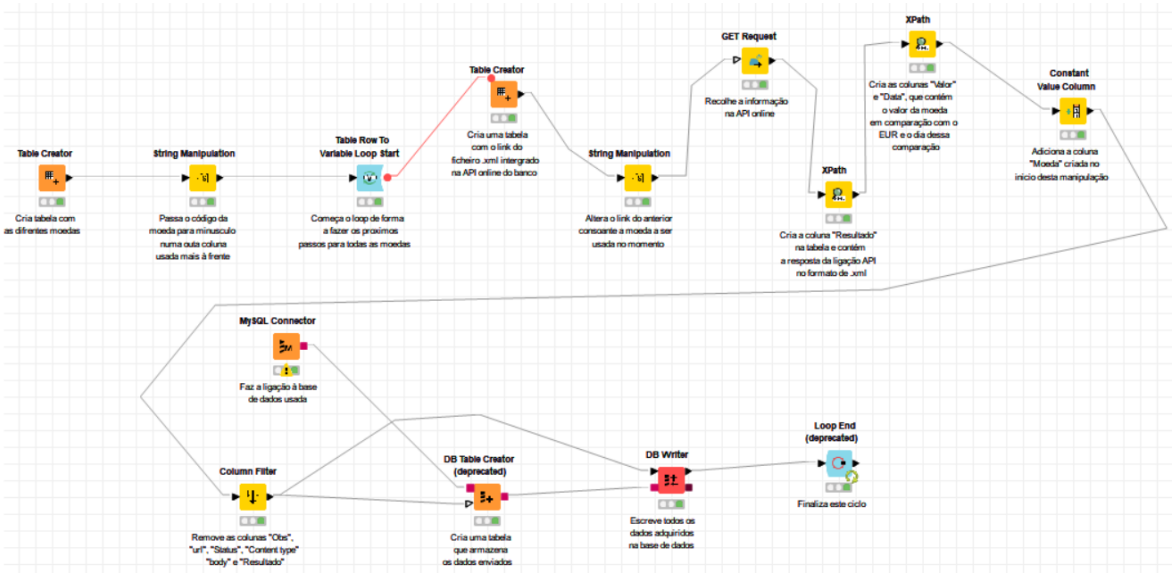


Figura 29: Terceiro workflow

O terceiro e último “workflow” é encarregue das operações mais complexas desde a ligação à API até à criação e registo da base de dados.

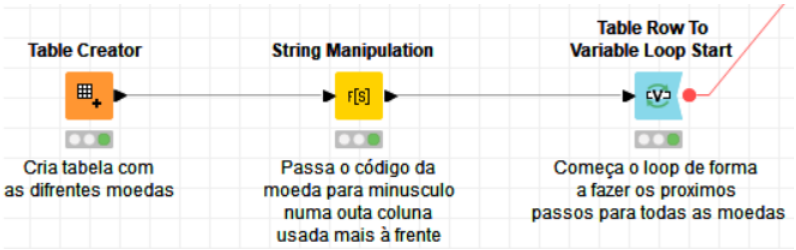
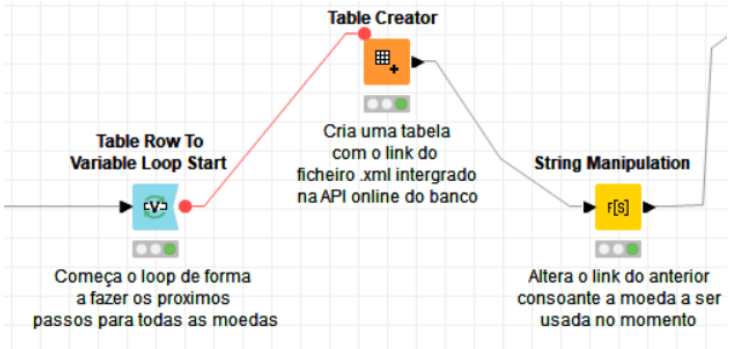


Figura 30: Início do workflow

Para iniciar criamos uma tabela com uma única coluna que contém o código de todas as moedas estrangeiras. Depois, usando a manipulação de strings passamos todas as moedas a letra minúscula e guardamo-las numa coluna nova que vai ser encarregue associar uma moeda à ligação à API em cada fase do ciclo criado já a seguir.



É iniciado um loop que vai permitir todos os próximos passos para cada moeda. Criamos uma tabela só com uma célula que contém o link da ligação ao ficheiro XML que contém os valores de cada moeda ao longo do tempo.

Figura 31: Criação da tabela com o link da ligação à API

Vamos usar a manipulação de strings outra vez de modo a poder substituir um excerto do link da ligação API com a variável da moeda a ser usada naquela fase do nosso ciclo.

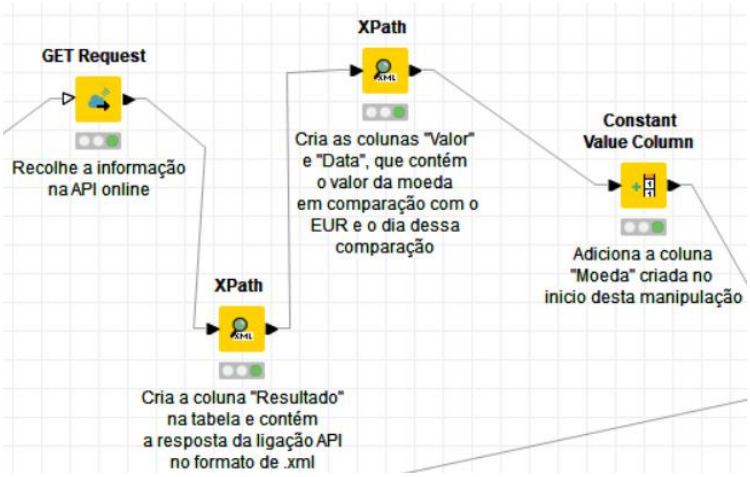


Figura 32: Método GET e XPath's

Segundo a ordem de tarefas, vamos fazer um método GET para recolher a informação online disponível na API do banco Central Europeu.

Os próximos dois nodes XPath vão recolher um determinado excerto do código XML dado pela API. Sendo que o primeiro node vai criar uma coluna chamada resultado que para cada célula vai representar a resposta da ligação perante o nosso pedido de acesso a uma determinada moeda e um determinado dia.

XPath summary

Column name	XPath query	Type
Resultado	/dns:CompactData/dns0:Dat...	Node(Multiple Rows)

Namespaces

Prefix	Namespace
dns	http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message
dns0	http://www.ecb.europa.eu/vocabulary/stats/exr/1

Add
Remove

Figura 33: Propriedades do primeiro XPath

O outro node vai ser encarregue de ir buscar à resposta feita em cima, o valor da moeda e a data em que se encontra. Com isto é mais fácil apresentar os dados na tabela final.

XPath summary

Column name	XPath query	Type
Valor	/dns:Obs/@OBS_VALUE	String(SingleCell)
Data	/dns:Obs/@TIME_PERIOD	String(SingleCell)

Namespaces

Prefix	Namespace
dns	http://www.ecb.europa.eu/vocabulary/stats/exr/1

Add
Remove

Figura 34: Propriedades do segundo XPath

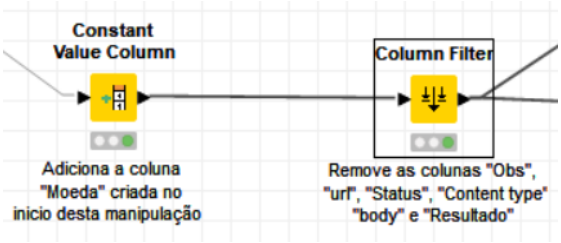


Figura 35: Remoção de colunas

De seguida vamos criar um node responsável por afixar a coluna “Moeda”, criado no início deste “worflow”, aonde vai ser guardada o código da moeda usada numa determinada fase do ciclo.

Continuando com o circuito, iremos remover as colunas de resposta do XML que não são relevantes para a demonstração de dados na tabela final.

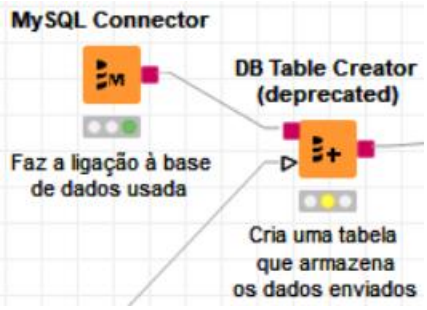


Figura 36: Ligação de base de dados e criação de tabela

Depois, será necessário fazer a ligação à nossa base de dados, aonde iremos guardar toda a informação recolhida até a este ponto. Consequentemente, teremos de criar uma tabela capaz de armazenar os dados enviados para a nossa base de dados.

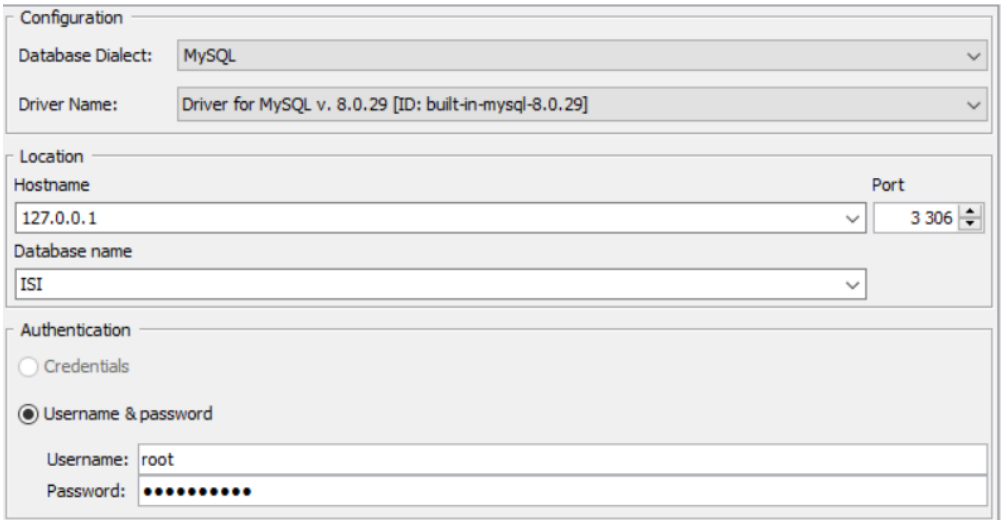


Figura 37: Ligação à base de dados

A ligação é feita escolhendo o tipo de base de dados, neste caso irá ser MySQL, escrevendo o “hostname”, a porta e o nome da mesma. A autenticação é feita de maneira tradicional, apenas é necessário digitar o nome de utilizador e a palavra-passe.

Para criar a tabela é bastante simples, basta escolher o “schema” aonde esta vai permanecer, atribuir um nome e indicar que colunas vão ser criadas.

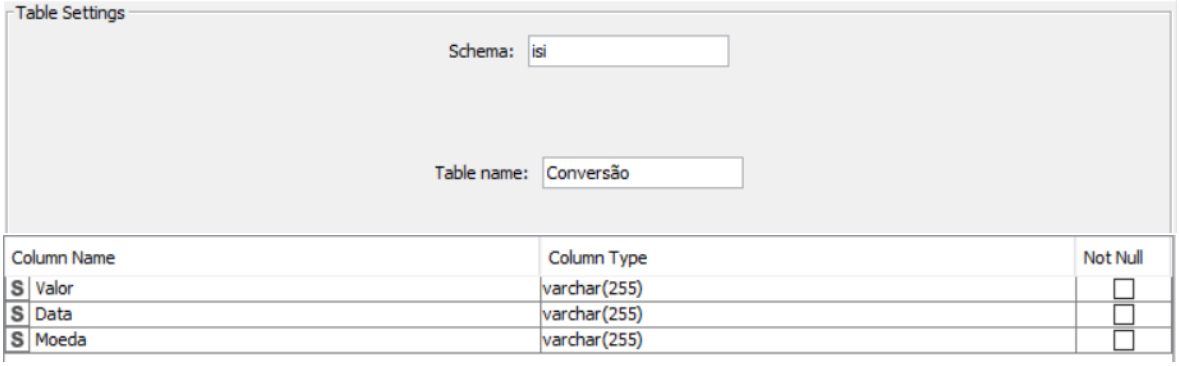


Figura 38: Criação da tabela

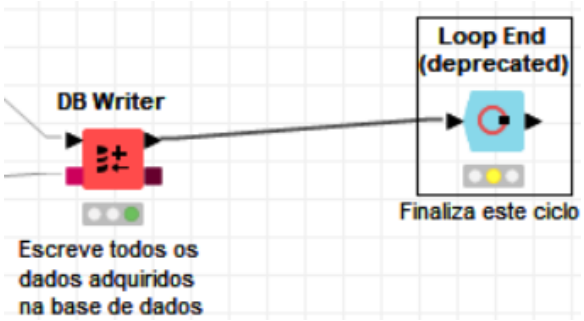


Figura 39. Final do ciclo

Para finalizar o nosso projeto, teremos de adicionar um node que vai selecionar a tabela criada anteriormente e vai escrever lá dentro toda a informação adquirida.

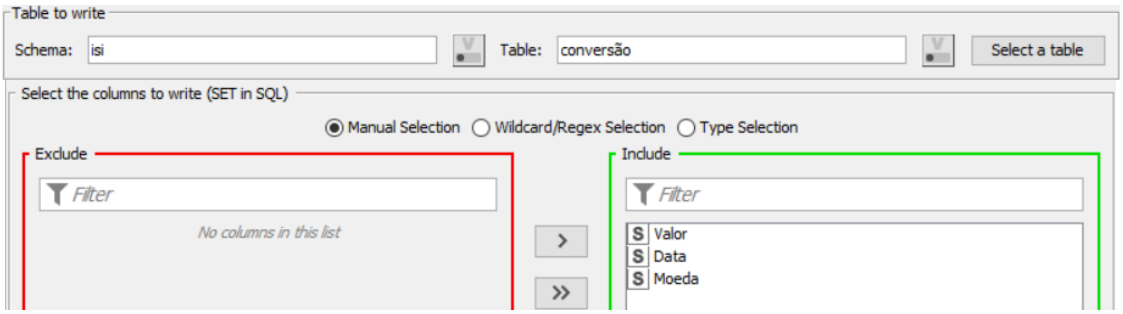


Figura 40: Escrita dos dados na tabela

O último node apenas vai ser encarregue de finalizar o ciclo quando todas as moedas forem convertidas para o Euro e já não haja mais nada a fazer.

5. Pentaho Kettle vs Knime

Como foi referido previamente, houve algumas diferenças na realização do projeto para cada elemento do grupo devido às diferenças que as duas plataformas possuem.

- **Isolação das transformações dos Jobs:** Algo que foi possível verificar logo no início do projeto, foi que o funcionamento do “Kettle” baseava-se na criação de transformações que iriam ser usadas futuramente nos Jobs, daí o utilizador ter de primeiro selecionar a aba de transformações para poder depois criar um JOB. Isso não acontece no “Knime” simplesmente temos um workflow aonde iremos fazer ambos as transformações como os Jobs.

- **Transformação de múltiplos ficheiros em simultâneo:** Como foi explicado em cima foi necessário criar um “workflow” exclusivo à segunda solução pois no “Knime” não era possível aplicar o mesmo filtro para os três ficheiros CSV ao mesmo tempo, foi necessário juntá-los para depois ser possível a transformação.

- **Processos mais atualizados:** Uma área em que o “Kettle” perde terreno é na atualidade dos processos existentes na plataforma, dado que muitos dos nodes que se encontram no leque de opções já não são relevantes para o dia de hoje. Para agravar isso, acontece o facto de não existirem nodes criados pela comunidade que ajudam para uma situação específica, algo que incentiva muito o desenvolvimento do projeto no “Knime”.

- **Debugging:** Outra das situações desagradáveis para um programador é quando um erro acontece e o programa não consegue indicar ao utilizador que erro é. Em parte, isto é o que acontece com o “Kettle” que dado um erro é listado um grande número de linhas na consola aonde apenas uma lá no meio realmente ajuda a corrigir o erro. Isto não acontece no “Knime”, se houver um erro a consola vai indicar especificamente aonde está o erro e porque este está a acontecer.

6. Vídeo de demonstração

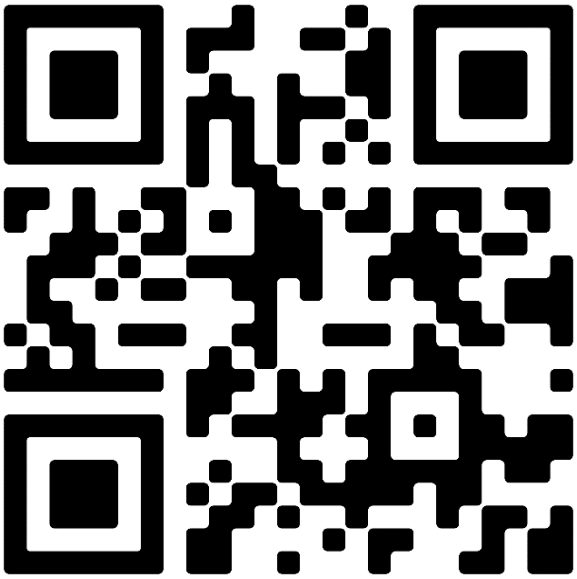


Figura 41: QR Code com vídeo de demonstração

7. Conclusão

No geral foi um trabalho bastante lúdico e intuitivo de se fazer. Isto acontece porque as plataformas usadas para desenvolver o projeto ajudam bastante na autonomia do utilizador devido à simplicidade de como se desenvolve cada tarefa.

No futuro se voltasse a fazer este trabalho se calhar tentava melhorar a otimização do projeto para poder usar o menor número de nodes para um maior número de tarefas. Isto pode-se verificar com a alta repetição de determinadas transformações.

8. Bibliografia

Plataforma usadas para a realização do projeto:

1. Knime

2. Pentaho Kettle

Complementos ao trabalho:

1. Knime Hub

2. Drona-hq