

Relatório do trabalho da disciplina de Integração de Sistemas de Informação

# Desenvolvimento de processos e aplicação de ferramentas de ETL

---

António Jorge Magalhães da Rocha – a26052

Licenciatura em Engenharia de Sistemas Informáticos (Pós-Laboral)



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.

António Jorge Magalhães da Rocha – a26052

## Índice

1	ENQUADRAMENTO	5
1.1	Contexto	5
1.2	Fundamento	5
2	PROBLEMA	6
3	SOLUÇÃO	6
4	ESTRATÉGIA DE DESENVOLVIMENTO	7
4.1	Contextualização	7
4.2	Escolha de software (perspetiva empresarial)	7
4.3	Escolha de software (perspetiva de desenvolvimento)	7
4.4	Decisão	8
4.5	Validação	8
4.6	Preparação do ambiente de desenvolvimento	8
5	WORKFLOW: CÓDIGOS POSTAIS	12
5.1	Códigos postais e Ruas	12
5.1.1	Extract Context Properties	12
5.1.2	String Manipulation	13
5.1.3	String to Path e Delete Files/Folders	14
5.1.4	Variable Creator	14
5.1.5	Script	15
5.2	Concelhos	16
5.2.1	Caminho local	17
5.2.2	Caminho completo	17
5.2.3	Avaliar (IF)	17
5.2.4	Apaga	18
5.2.5	Download	18
5.2.6	Case Switch Start	19
5.2.7	E-mail sucesso	20
5.2.8	E-mail falha	21
5.2.9	Case Switch End	21

5.2.10	Table Row to Variable e CSV Reader	21
5.3	Distritos	22
5.3.1	Webpage Retriever	22
5.3.2	XPath	22
5.3.3	Cell Splitter	23
5.3.4	Unpivot	23
5.3.5	Cell Splitter (2)	24
5.4	Cruzar tabelas: Códigos Postais e Ruas, Concelhos, Distritos	24
5.4.1	Value Lookup	25
5.4.2	Joiner	25
5.5	Limpar	26
5.6	Controlo de erros	27
5.7	Load	28
5.7.1	Escrever na Base de Dados	28
5.7.2	Query à Base de dados	29
5.8	Agregar, analisar e visualizar	29
6	WORKFLOW EXTRA A: API CONDIÇÕES RODOVIÁRIAS – ALERTA E-MAIL	32
7	WORKFLOW EXTRA B: GRUPOS ETÁRIOS .TABLE	32
8	WORKFLOW EXTRA C: RECEITA SOPA – GOOGLE DRIVE E REGEX	33
9	CASOS DE USO / EXPERIÊNCIAS / CONSIDERAÇÕES	34
9.1	Base de dados SQL	34
9.1.1	Conexão: PostgreSQL e MySQL	34
9.1.2	Autenticação e segurança	35
9.1.3	Execução de queries	36
9.1.4	Mapeamento de tipos de dados	36
9.1.5	Reconexão automática a bases de dados	37
9.2	Sharepoint	37
9.3	Knime Community Hub	38
9.4	Codificação de caracteres	38
9.4.1	Caracteres especiais	38
9.4.2	Leading Zeros	39

9.5	Abstração, representação e documentação gráfica do processo-----	40
9.6	Metanodes de Metanodes-----	40
9.7	GroupBy : Categorização e agregação -----	41
9.8	Ler e/ou juntar ficheiros CSV-----	41
9.9	Identificação de colunas-----	42
10	VÍDEO COM DEMONSTRAÇÃO (QR CODE) -----	43
11	CONCLUSÃO E TRABALHOS FUTUROS-----	45
12	REFERÊNCIAS BIBLIOGRÁFICAS-----	47
13	ANEXO A – CERTIFICADO “BASIC PROFICIENCY IN KNIME ANALYTICS PLATFORM”	48

# 1 Enquadramento

## 1.1 Contexto

Este trabalho prático insere-se no âmbito da Unidade Curricular Integração de Sistemas de Informação, do curso de Licenciatura em Engenharia de Sistemas Informáticos, lecionado no Instituto Politécnico do Cávado e do Ave pelo Professor Luís Ferreira. É autoria do aluno António Jorge Magalhães da Rocha, sob a mentoria do Professor responsável.

O objetivo principal do trabalho é proporcionar uma experiência de desenvolvimento de soluções recorrentemente requisitadas e essenciais: soluções de ETL (extract, transform, load). Não só para o funcionamento e integração de sistemas informáticos, mas também para a tomada de decisões informadas por parte de indivíduos e organizações.

Permitirá ainda recorrer, consolidar e expandir o conhecimento adquirido em outras unidades curriculares e/ou na esfera pessoal dos alunos nomeadamente nos domínios de gestão de projetos de engenharia de software, bases de dados, estruturas e manipulação de dados e programação.

## 1.2 Fundamento

Os processos de ETL endereçam uma dificuldade atual e recorrente no tecido empresarial: a existência, por si só, dos dados não traz conhecimento para a organização.

Para que os dados possam ser uma mais valia é necessário que sejam obtidos, avaliados, validados, integrados de forma relacional, transformados e, por fim, analisados. Para que esta tarefa possa ser feita com facilidade, elevada frequência, baixa intervenção humana e com garantia de fiabilidade, é necessário que seja automatizada. É neste contexto que os processos de ETL podem suportar, com excelência, as organizações e configuram uma solução de elevada qualidade e sustentabilidade.

A solução será desenvolvida de forma que, a partir da informação, possa ser gerado conhecimento efetivo que acrescente valor. Este conhecimento do seu mercado e dos seus clientes e, em última análise, de si própria, é essencial para a organização. Qualquer estratégia organizacional atual ou futura será beneficiária desta informação e do conhecimento a partir dela obtido.

## 2 Problema

O objeto de estudo do trabalho é uma empresa de entrega de vendas online.

Esta empresa dedica consideráveis recursos para obter informação essencial que poderia ser automatizada com relativa facilidade e retorno. A situação atual, baseada na ação humana manual e repetitiva, para além de diminuir a produtividade a frequência de atualização, aumenta a complexidade do acesso à informação e a probabilidade de erros.

Para além disto, os colaboradores são forçados a recorrer a diferentes bases de dados, tanto internas e externas. A dispersão e falta de normalização dos dados limita o benefício de conhecimento que poderia ser obtido caso existisse um processo robusto de ETL implementado.

Frequentemente, a existência destes dados em plataformas/sistemas diferentes aumenta, aos olhos das organizações, a complexidade que um eventual esforço de integração automatizada iria representar deixando, por vezes, estes projetos pendentes.

## 3 Solução

Disposta a abrir a porta a processos automáticos de ETL, a empresa contactou a consultoria a26052, que detém o certificado [Anexo A – Certificado “Basic Proficiency in KNIME Analytics Platform”](#).

Para avaliar o potencial transformativo que os processos de ETL poderão ter, a empresa definiu os seguintes objetivos:

- Testar e demonstrar a capacidade de aceder, com segurança e as credenciais adequadas a cada utilizador, e em qualquer máquina, a todas as fontes de informações utilizadas pela empresa.
- Montar um processo de ETL que permita extrair dados oficiais de códigos postais e moradas, transformá-los, e carregá-los numa única base de dados PostgreSQL.
- Montar um processo de ETL que emita avisos com base nas condições climáticas.
- Despoletar avisos por e-mail do sucesso ou falha do processo de ETL.
- Demonstrar a possibilidade de implementar análise gráfica automática em formato de *dashboard*.
- Avaliar a qualidade dos dados e reportar dados não conformes/incompletos.
- Demonstrar a capacidade da ferramenta no que toca a manipulação do sistema de ficheiros do sistema operativo local.
- Demonstrar a capacidade da ferramenta de comunicar visualmente o processo de ETL e auxiliar a sua documentação, nomeadamente no que toca a diferentes níveis de abstração.

## 4 Estratégia de desenvolvimento

A estratégia de trabalho para o desenvolvimento da solução consiste numa abordagem metódica e faseada, pretendendo garantir um entendimento correto e boas decisões em cada passo antes de prosseguir. Pretende também não deixar nenhuma solução viável por avaliar.

### 4.1 Contextualização

- Conhecimento das necessidades
- Conhecimento das dificuldades técnicas e operacionais
- Sistemas de bases de dados e/ou fontes de informação
- Já existe algum processo, ferramenta ou sistema de ETL em utilização?
- Utilizadores finais (quem são? Quão a sua literacia informática?)
- Sistema de autorização de fonte de dados (OAuth etc)
- Nível autorização de cada utilizador em cada sistema (tem acesso aos clientes, incluindo ao seu nome? Tem acesso às vendas? Tem acesso aos valores envolvidos?)

### 4.2 Escolha de software (perspetiva empresarial)

- Que validações será necessário fazer aos dados?
- Qual o nível de escalabilidade necessário?
- Qual o nível de esforço de que pretende dedicar à manutenção e atualização da solução?
- Pretende uma solução totalmente personalizada (custom scripts etc) ou aceita que inclua ferramentas third party?
- Caso aceite que seja incluída uma third party, aceita uma solução closed source?
- Aceita uma solução open source?
- Aceita que sejam incluídas ferramentas que irão gerar custos para a empresa?
- Pretende analisar ou visualizar os dados em alguma plataforma específica?

### 4.3 Escolha de software (perspetiva de desenvolvimento)

- Que ferramentas se enquadram no contexto, dificuldades e requisitos e que permitem atingir o objetivo?
- Que ferramentas permitem a extração dos dados da plataforma onde se encontram?
- Que ferramentas necessitam da configuração de processos intermédios para a extração dos dados e quais o conseguem fazer de forma integrada?
- Que ferramentas são capazes de transformar os dados da forma pretendida?
- Alguma das ferramentas tem a capacidade integrada necessária para a análise de dados da forma que a organização pretende, sem recorrer a exportação de dados?
- A organização tem o nível de certeza necessária de que as suas necessidades de análise de informação não irão ultrapassar o que a ferramenta proporciona atualmente?
- Que ferramentas têm a capacidade de representar os dados para análise da forma que a organização pretende ou poderá pretender no futuro?
- Existe alguma ferramenta única que consiga realizar todas as etapas do processo com a mesma qualidade que uma solução que recorra a várias ferramentas?



## 4.4 Decisão

A ferramenta escolhida para cumprir os objetivos e requisitos foi o Knime Analytics Platform.

É um software grátis (exceto a componente server), open source, altamente escalável, com um potencial enorme de modularidade e um suporte robusto, graças a uma documentação de qualidade e a uma comunidade. Apesar do dinamismo e velocidade de crescimento deste software, é evidente o compromisso com a compatibilidade, que é evidenciado pelo support aos “nodes” designados como “legacy”, que fazem com que qualquer workflow desenhado na última década continue a ser válido e funcional.

Permite também a melhoria contínua em equipa graças ao Knime Community Hub, que permite sincronização dos workflows e definição de repositórios privados e públicos. Caso a empresa atinga um nível de processos ETL que o justifique, poderá também optar por agendar os jobs através do Knime Server, subescrevendo o mesmo.

No entanto, e acima de tudo, a decisão só poderá recair sobre este software se o mesmo permitir atingir facilmente, e com clareza, todos os objetivos, tanto a curto como a médio prazo da empresa.

## 4.5 Validação

Para validar que a ferramenta tem, de facto, capacidade para dar resposta a todos os objetivos, foram realizados testes curtos e específicos. Estes testes podem dividir-se em três categorias:

- testes a funções essenciais para a implementação do processo de ETL dos códigos postais;
- testes que fazem parte do que a empresa requisitou, já que serão necessários em eventuais futuras implementações de novos processos ETL;
- testes realizados porque, apesar de não requisitados, serem provavelmente necessários no futuro.

A ferramenta provou ser capaz de realizar os objetivos. É de salientar o elevado grau de flexibilidade na forma como o resultado pretendido pode ser atingido.

## 4.6 Preparação do ambiente de desenvolvimento

Para configurar o ambiente de desenvolvimento foi necessário:

- Instalar o Knime e vários pacotes de extensões, PostgreSQL, MySQL, Conda, Python, bibliotecas Python e um IDE.
- Criar base de dados populadas, pastas e ficheiros de teste no sistema de ficheiros local e em várias clouds.
- Criar repositórios de workflows no Knime Community Hub.

De todos estes passos, o mais importante a salientar é a configuração do ambiente Python e das suas bibliotecas de forma a poder integrar scripts Python no workflow.

Uma das características que tornam o Knime tão poderoso é a capacidade utilizar linguagens de programação embutidas no workflow, tendo um suporte muito extenso de linguagens e bibliotecas/frameworks.

Para atingir o objetivo principal da empresa, a implementação de um processo ETL que sintetize e mantenha atualizada uma base de dados única de moradas e códigos postais, é essencial utilizar manipulação de páginas web. Isto deve-se ao facto de as fontes oficiais apenas disponibilizarem os dados brutos através de download e páginas web que requerem autenticação.

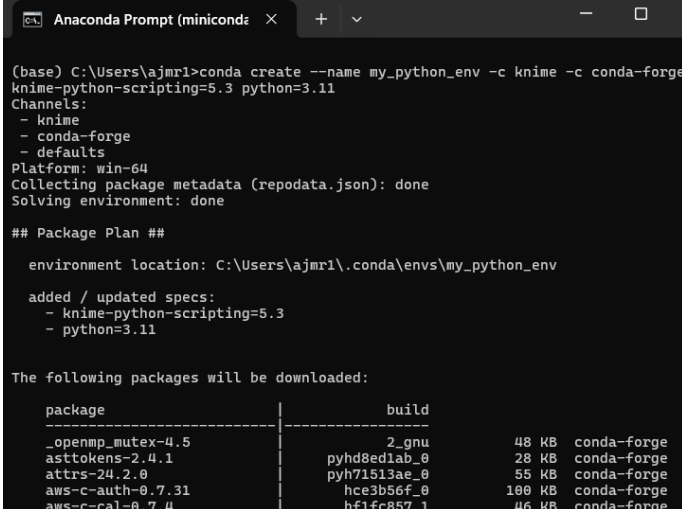
Uma das melhores bibliotecas para o fazer é a Selenium, para Python, através do Webdriver.

Existem pacotes de nodes para o Knime que permitem utilizar funcionalidades do Selenium sem desenvolver código. No entanto, esses pacotes são pagos.

Existe, no entanto, uma forma de utilizar Selenium em todo o seu potencial sem quaisquer custos: implementar um ambiente Python dedicado para o workflow, com o pacote Selenium instalado. O Knime suporta a utilização de ambientes Python de três formas diferentes. Neste workflow a integração será por Conda.

O processo de configuração seguido foi o seguinte:

1. Preparar o environment “my\_python\_env” e instalar os packages necessários para o integrar no knime



```
(base) C:\Users\ajmr1>conda create --name my_python_env -c knime -c conda-forge
knime-python-scripting=5.3 python=3.11
Channels:
 - knime
 - conda-forge
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

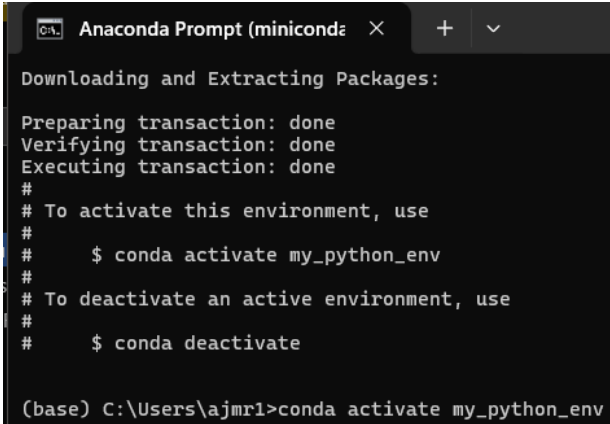
environment location: C:\Users\ajmr1\.conda\envs\my_python_env

added / updated specs:
 - knime-python-scripting=5.3
 - python=3.11

The following packages will be downloaded:
```

package	build	size	channel
_openmp_mutex-4.5	2_gnu	48 KB	conda-forge
asttokens-2.4.1	pyhd8ed1ab_0	28 KB	conda-forge
attrs-24.2.0	pyh71513ae_0	55 KB	conda-forge
aws-c-auth-0.7.31	hce3b56f_0	100 KB	conda-forge
aws-c-cal-0.7.4	hf1fc857_1	46 KB	conda-forge

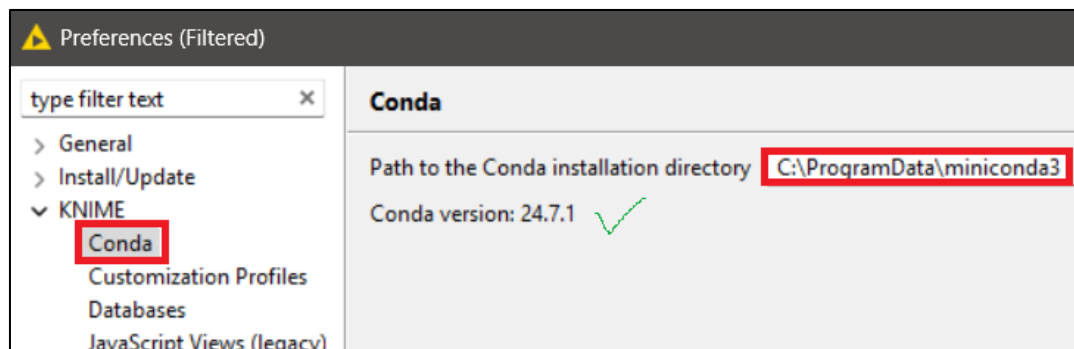
2. Activar o ambiente



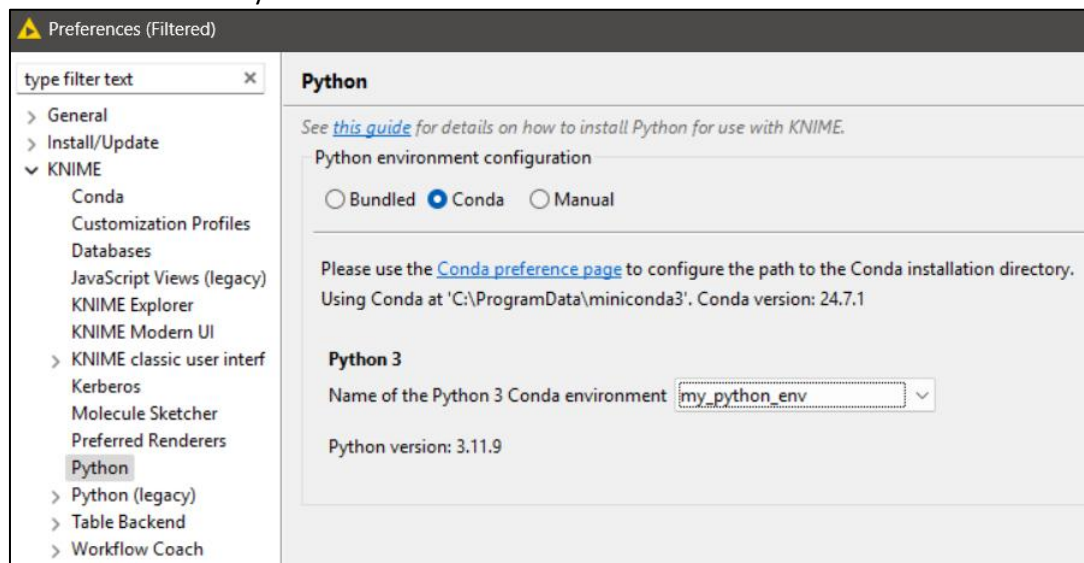
```
Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate my_python_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\Users\ajmr1>conda activate my_python_env
```

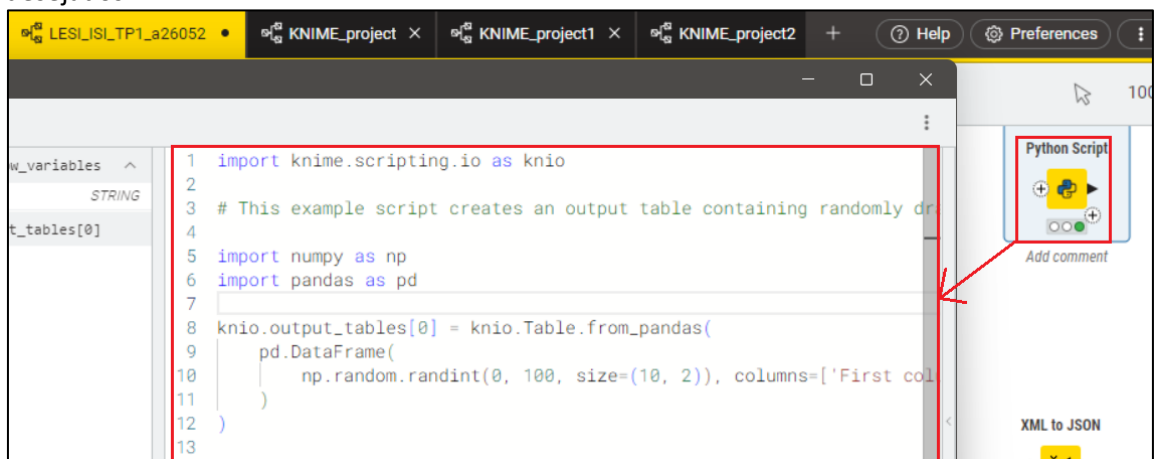
3. Definir o diretório Conda



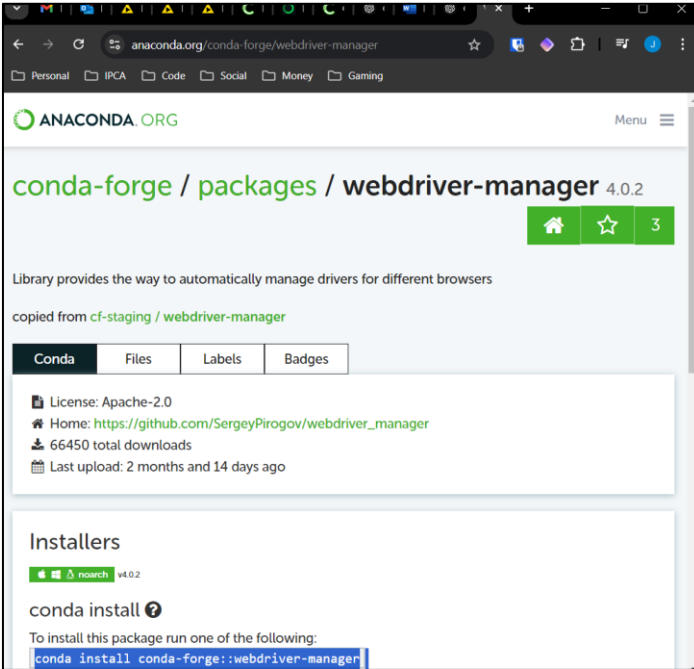
#### 4. Definir o ambiente Python



#### 5. Já é possível utilizar nodes customizados a correr código Python, com quaisquer pacotes desejados.



6. Instalar o webdriver a partir das sources do Conda



conda-forge / packages / webdriver-manager 4.0.2

Library provides the way to automatically manage drivers for different browsers

copied from cf-staging / webdriver-manager

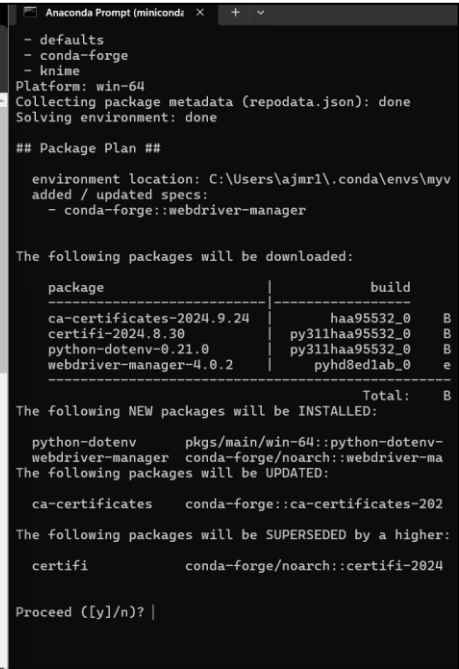
Conda Files Labels Badges

License: Apache-2.0  
Home: [https://github.com/SergeyPirogov/webdriver\\_manager](https://github.com/SergeyPirogov/webdriver_manager)  
66450 total downloads  
Last upload: 2 months and 14 days ago

Installers

conda install

To install this package run one of the following:  
conda install conda-forge::webdriver-manager



```
- defaults
- conda-forge
- knime
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\ajmr1\.conda\envs\myv
added / updated specs:
- conda-forge::webdriver-manager

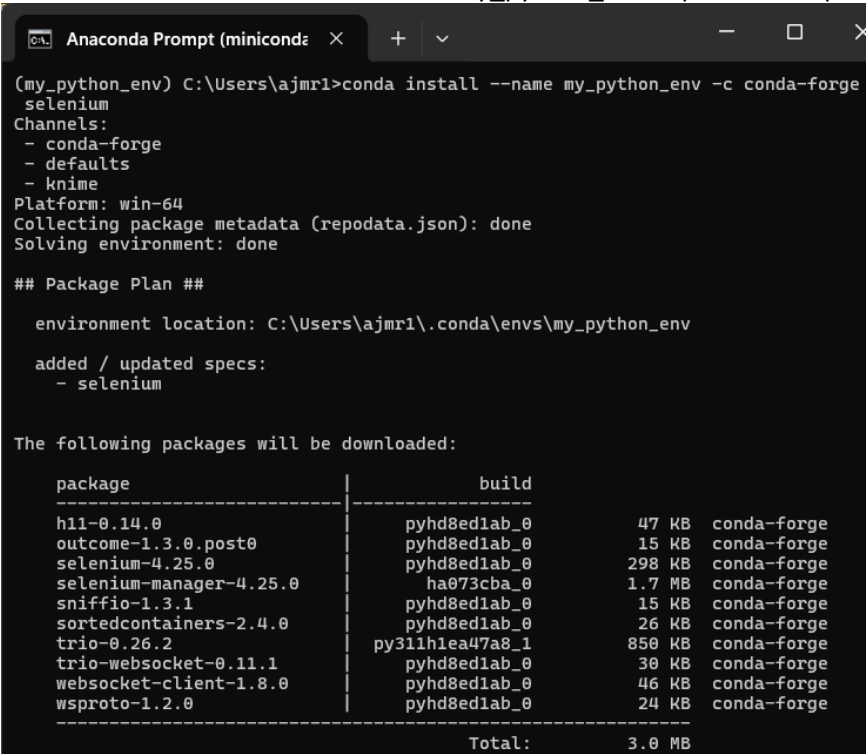
The following packages will be downloaded:

package | build
-----|-----
ca-certificates-2024.9.24 | haa95532_0 B
certifi-2024.8.30 | py311haa95532_0 B
python-dotenv-0.21.0 | py311haa95532_0 B
webdriver-manager-4.0.2 | pyhd8ed1ab_0 e
Total: B

The following NEW packages will be INSTALLED:
python-dotenv pkgs/main/win-64::python-dotenv-
webdriver-manager conda-forge/noarch::webdriver-ma
The following packages will be UPDATED:
ca-certificates conda-forge::ca-certificates-202
The following packages will be SUPERSEDED by a higher:
certifi conda-forge/noarch::certifi-2024

Proceed ([y]/n)?
```

7. Instalar o Selenium no environment my\_python\_env a partir do repositório conda-forge



```
(my_python_env) C:\Users\ajmr1>conda install --name my_python_env -c conda-forge selenium
Channels:
- conda-forge
- defaults
- knime
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\ajmr1\.conda\envs\my_python_env

added / updated specs:
- selenium

The following packages will be downloaded:

package | build | size | channel
-----|-----|-----|-----
h11-0.14.0 | pyhd8ed1ab_0 | 47 KB | conda-forge
outcome-1.3.0.post0 | pyhd8ed1ab_0 | 15 KB | conda-forge
selenium-4.25.0 | pyhd8ed1ab_0 | 298 KB | conda-forge
selenium-manager-4.25.0 | ha073cba_0 | 1.7 MB | conda-forge
sniffio-1.3.1 | pyhd8ed1ab_0 | 15 KB | conda-forge
sortedcontainers-2.4.0 | pyhd8ed1ab_0 | 26 KB | conda-forge
trio-0.26.2 | py311h1ea47a8_1 | 850 KB | conda-forge
trio-websocket-0.11.1 | pyhd8ed1ab_0 | 30 KB | conda-forge
websocket-client-1.8.0 | pyhd8ed1ab_0 | 46 KB | conda-forge
wsproto-1.2.0 | pyhd8ed1ab_0 | 24 KB | conda-forge
Total: 3.0 MB
```

## 5 WORKFLOW: Códigos Postais

O *workflow* principal pretende criar uma base de dados centralizada e atualizável de forma automática.

As fontes de informação necessárias são as seguintes:

- Uma lista de códigos de distrito e nomes de distrito.
- Uma lista de códigos de distrito e concelho e nomes dos concelhos.
  - O código de concelho não é único, apenas quando conjugado com o código de distrito.
- Uma lista de moradas e códigos postais com códigos de distrito e de concelho.

Os passos essenciais são os seguintes:

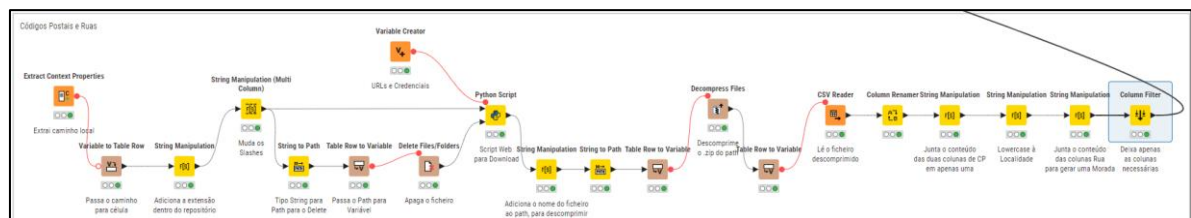
- Aceder às fontes de informação.
- Manipular informação.
- Validar informação.
- Detetar erros.
- Agregar informação
- Preparar e carregar a informação no seu destino final.
- Despoletar o envio de e-mails que indicam o sucesso ou falha do processo de ETL.

O objetivo final:

- Obter uma base dados PostgreSQL com uma tabela de moradas completas.
- Avaliar a qualidade da informação, nomeadamente quais e quantas moradas incompletas foram excluídas e que percentagem representam das moradas iniciais.

### 5.1 Códigos postais e Ruas

O ramo Códigos postais e Ruas pretende obter uma tabela de todos os códigos postais em Portugal, com a(s) rua(s) correspondente(s).

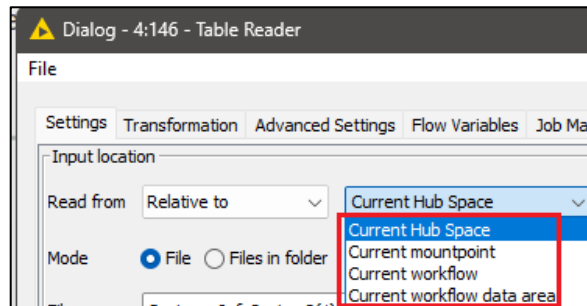


#### 5.1.1 Extract Context Properties

A utilização de caminhos relativos é crucial na garantia de funcionamento agnóstico à máquina local e utilizador que executam o processo de ETL.

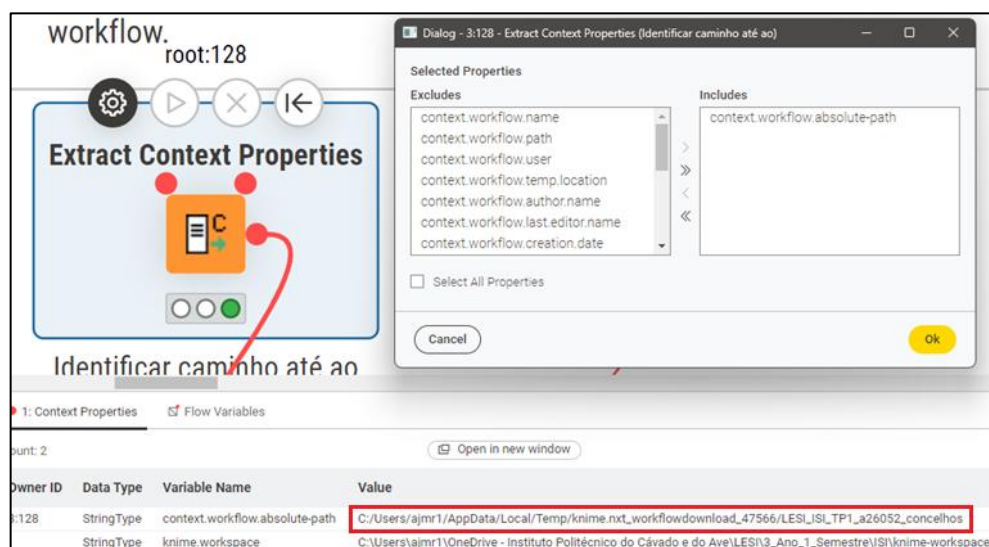
Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).

É também o que permite que os processos possam, pelo menos no caso do Knime, ser desenvolvidos e alojados permanentemente na cloud.



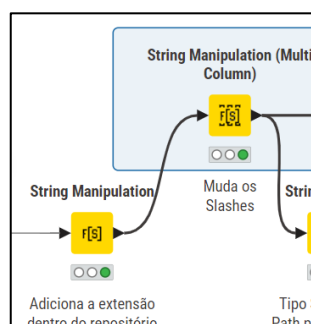
Quando um workflow alojado na cloud Knime Community Hub é descarregado para ser utilizado, é criado um diretório temporário para o alojar na máquina local. Este diretório tem uma componente aleatória, pelo que sem caminhos relativos seria impossível este tipo de colaboração/alojamento.

A imagem seguinte demonstra o node que permite obter inúmeras informações sobre o funcionamento local onde o workflow está a decorrer, algumas opções de caminhos relativos e também o output do caminho em vigor, assim como a sua componente aleatória “47566”.



### 5.1.2 String Manipulation

De seguida são utilizados dois nodes de “String Manipulation”, um deles específico a colunas mas não seria necessário.



Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).

O primeiro recebe o caminho:

```
"C:/Users/ajmr1/AppData/Local/Temp/knime.nxt_workflowdownload_91545/LESI_ISI_TP1_a26052_concelhos"
```

E adiciona o sufixo `"/data/zip_database_folder"`, que é o folder local que será utilizado e estará sempre disponível em qualquer máquina.

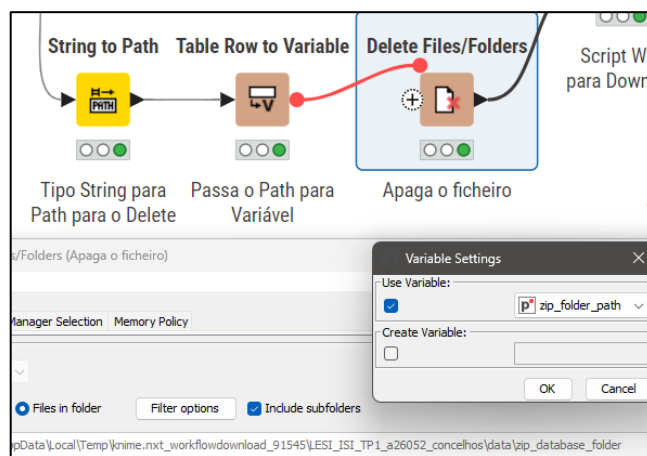
O segundo altera os slashes para um formato que possa ser utilizado pela script e pelo node `"Delete Files/Folders"`

```
1 replace($context.workflow.absolute-path$/, "/", "\\")
```

### 5.1.3 String to Path e Delete Files/Folders

Antes de realizar o download de um novo ficheiro, é necessário garantir que o antigo já não se encontra na mesma localização, o que poderia gerar conflitos e até a convicção de que a atualização teria ocorrido quando na verdade se estaria a utilizar uma versão antiga do ficheiro.

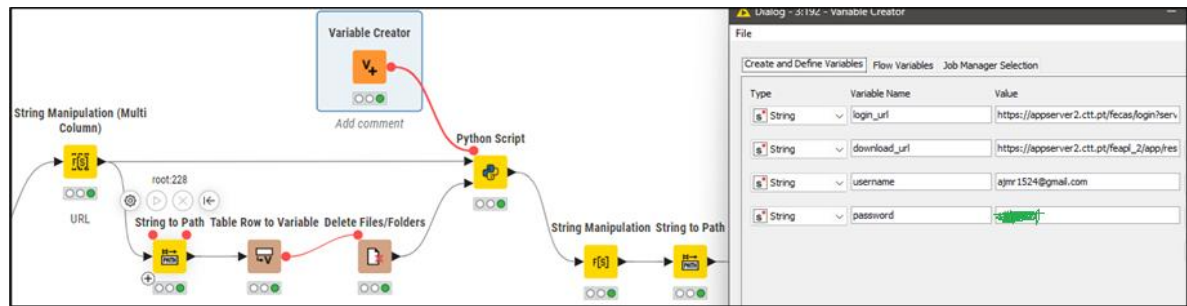
O caminho anterior é convertido de string para Path, e de célula para variável. Por fim, é passado ao node `"Delete Files/Folders"` que executa o processo de apagar o ficheiro através do caminho que lhe é passado.



### 5.1.4 Variable Creator

O node `"Variable Creator"` define os valores de quatro variáveis que serão passadas à script. Esta configuração torna-se mais clara e simples do que optar pelo `"hardcode"` destes parâmetros.

Estas variáveis são os URL's para autenticação no site dos CTT e para o download do ficheiro, assim como o username e password.



### 5.1.5 Script

A utilização de scripts customizadas abre um leque muito vasto de possibilidades aos processos de ETL. Principalmente quando é possível recorrer a qualquer *library* ou *framework*.

Nos exemplos seguintes, verificamos a utilização da biblioteca Selenium e do Webdriver para conseguir aceder e manipular páginas web através de tags (*id*, *element*, *label* etc), passar credenciais de autenticação de uma forma segura e, por fim, definir propriedades do download de ficheiros.

No caso do Knime, a integração destas scripts é feita de forma fluida e dá continuidade ao processo de *workflow* com bastante versatilidade, não interrompendo ou prejudicando o fluxo de informação.

No que toca ao conteúdo da script, podemos verificar, no painel esquerdo, que a mesma tem acesso às variáveis e tabelas que lhe são passadas.

A script começa por criar e popular variáveis relativas a caminhos e credenciais, configura o webdriver, navega até à página de autenticação, rejeita cookies, preenche o login e username, submete o formulário e, estando já autenticado, acede agora sim ao link para download. O ficheiro descarregado será guardado no caminho descrito pela variável "download\_path", graças à configuração do webdriver para tal.



Python Script

► Input Table 1

knio.input\_tables[0].to\_pandas()

context.workflow.absolute-path

String

► Input Table 2

knio.input\_tables[1].to\_pandas()

Path

Path

● Flow variables

knio.flow\_variables

login\_url

STRING

download\_url

STRING

username

STRING

password

STRING

knime.workspace

STRING

RowID

STRING

context.workflow.absolute-path

STRING

► Output Table 1

knio.output\_tables[0]

```

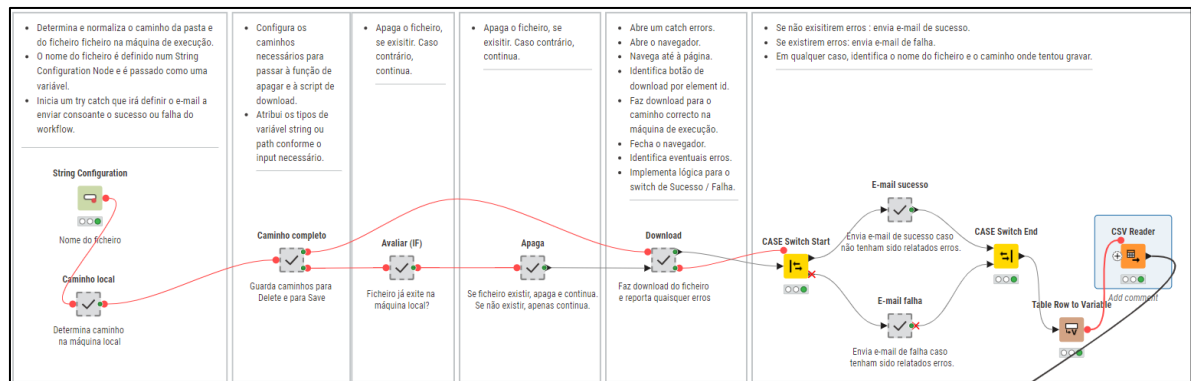
12 import pandas as pd # para o dataframe
13
14 # Define nome de utilizador, password e diretório de download
15 username = knio.flow_variables["username"]
16 password = knio.flow_variables["password"]
17 download_url = knio.flow_variables["download_url"]
18
19 df = knio.input_tables[0][context.workflow.absolute-path].to_pandas()
20 download_path = df.iloc[0][context.workflow.absolute-path]
21
22 # Configurar opções do Chrome
23 chrome_options = Options()
24 #chrome_options.add_argument("--headless") # Executar em modo headless, remova se quiser ver o navegador
25 chrome_options.add_argument("--disable-gpu")
26 chrome_options.add_experimental_option("prefs", {
27     "download.default_directory": download_path, # Definir o diretório de download
28     "download.prompt_for_download": False, # Desativar prompt de download
29     "download.directory_upgrade": True,
30     "safebrowsing.enabled": True
31 })
32
33 # Configurar o WebDriver
34 driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=chrome_options)
35
36 driver.get(knio.flow_variables["login_url"]) # vai à página de autenticação
37
38 # Espera pelo pop dos cookies e rejeita-os, caso apareça
39 try:
40     cookies_button = WebDriverWait(driver, 4).until(
41         EC.element_to_be_clickable((By.ID, "onetrust-reject-all-handler"))
42     )
43     cookies_button.click() # Clicar no botão
44
45 except TimeoutException:
46     print("O botão de cookies não apareceu a tempo.")
47
48 # Preencher o nome de utilizador
49 username_field = driver.find_element(By.ID, "username")
50 username_field.send_keys(username)
51
52 # Preencher a password
53 password_field = driver.find_element(By.ID, "password")
54 password_field.send_keys(password)
55
56 # Clicar no botão de login
57 login_button = driver.find_element(By.NAME, "btnSubmit")
58 login_button.click()
59
60 # Esperar pelo login
61 time.sleep(1)
62
63 # Vai à página de download
64 download_url = knio.flow_variables["download_url"]
65 driver.get(download_url)
66
67 # Espera pelo download
68 time.sleep(6)
69
70 # Fechar o driver
71 driver.quit()
72
73 # Passar diretamente a tabela de entrada para a tabela de saída sem conversão
74 knio.output_tables[0] = knio.input_tables[0] # Passar a segunda tabela de entrada como saída

```

## 5.2 Concelhos

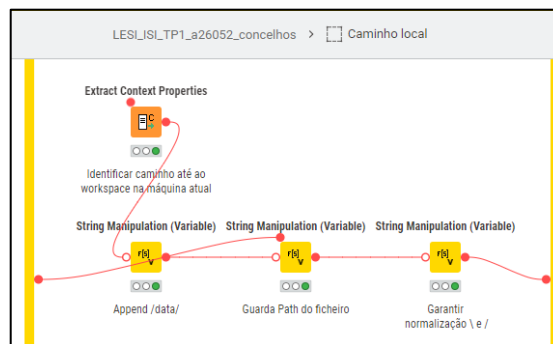
O objetivo do ramo Concelhos é obter uma tabela com o Código do Distrito, Código de Concelho (que não é único, apenas quando combinado com o Código do Distrito, daí a inclusão desta coluna neste ramo) e, por fim, do Nome do Concelho.

Devido à extensão deste ramo e para fins de experimentação foram utilizados Metanodes e Metanodes compostos por Metanodes.



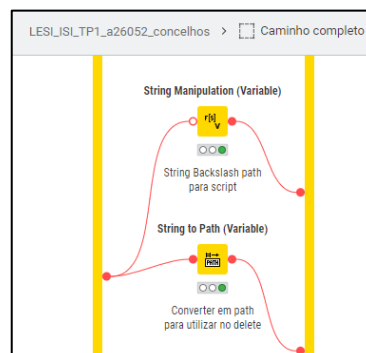
### 5.2.1 Caminho local

O *metanode* “Caminho local” trata de identificar o caminho correcto na máquina local (mais sobre isto no capítulo 5.1.1), de fazer append a sufixos necessários para gerar o caminho pretendido e de normalizar o uso de *slashes* nos caminhos.



### 5.2.2 Caminho completo

O *metanode* “Caminho completo” tem como função guardar em variáveis os caminhos prontos a serem utilizados na script python e também no node “Delete Files/Folders”.

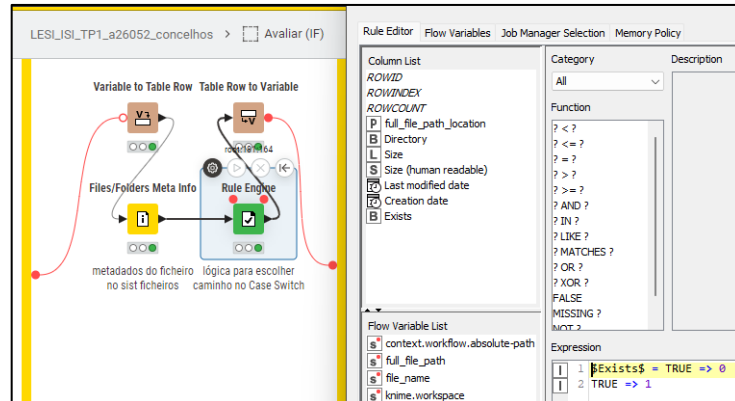


### 5.2.3 Avaliar (IF)

No *metanode* “Avaliar (IF)” é de salientar:

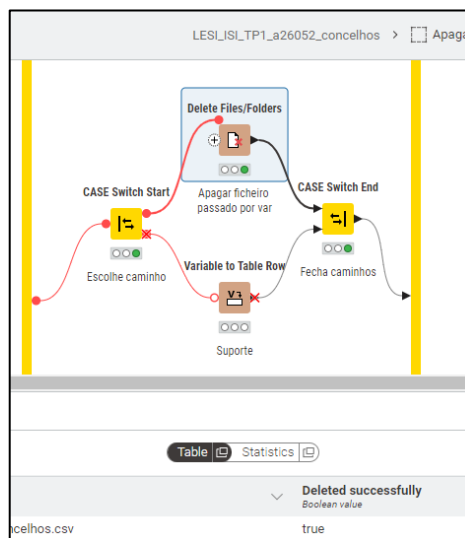
Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).

- o *node* “File/Folders Meta Info” que retira informações sobre o ficheiro alvo no sistema de ficheiros local, nomeadamente se existe ou não, que é o objetivo neste passo.
- o *node* “Rule Engine” que, com base na coluna “Exists” ser verdade ou não, determina a execução da porta 0 ou 1 no “Case Switch”, mais à frente e fora deste *metanode*.



### 5.2.4 Apaga

Dentro do *metanode* “Apaga”, caso o Case Switch receba a instrução (do Rule Engine) de que deve executar a porta 0 (porque o ficheiro existe), irá apagar o ficheiro. Caso execute a porta 1 (porque o ficheiro não existe, não é realizada qualquer ação a não ser converter o output para tabela e poder fechar o Case Switch.



### 5.2.5 Download

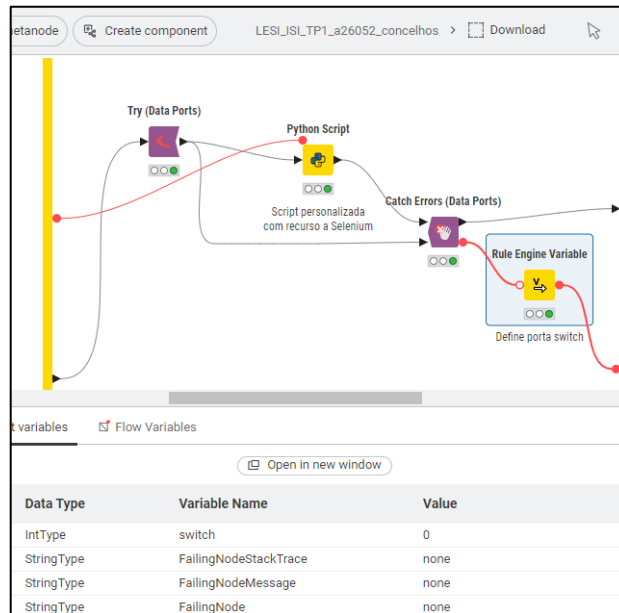
O *metanode* “Download” é constituído por um Try-Catch que envolve uma script python feita de forma customizada.

Estes *nodes* de Try-Catch reconhecem erros e guardam propriedades sobre os mesmos em variáveis (ex: FailingNodeStackTrace, FailingNodeMessage, FailingNode) passando essa informação ao restante fluxo. Estas variáveis são visíveis na imagem seguinte.

Neste caso, essa informação será crucial para determinar o conteúdo do aviso que será enviado por e-mail mais à frente.

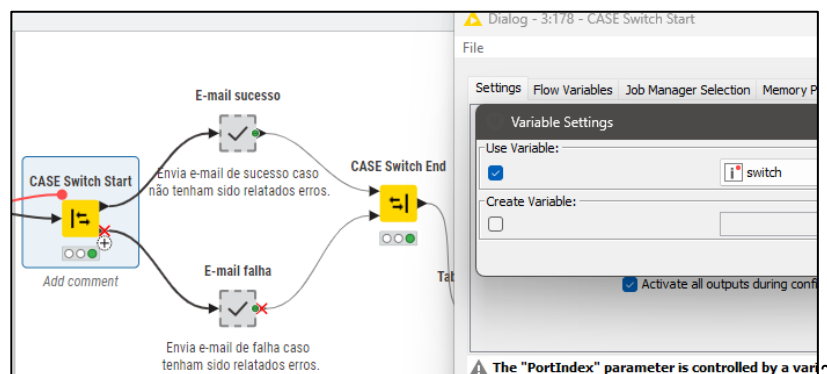
De notar que as informações em formato tabela fluem através do Try-Catch, tal como as variáveis de controlo de erro.

Consoante o output do try catch, o “Rule Engine Variable” irá determina a porta, graças à variável “switch” visível na imagem, a porta a executar no Case Switch seguinte. No caso exemplificado, será a porta 0.



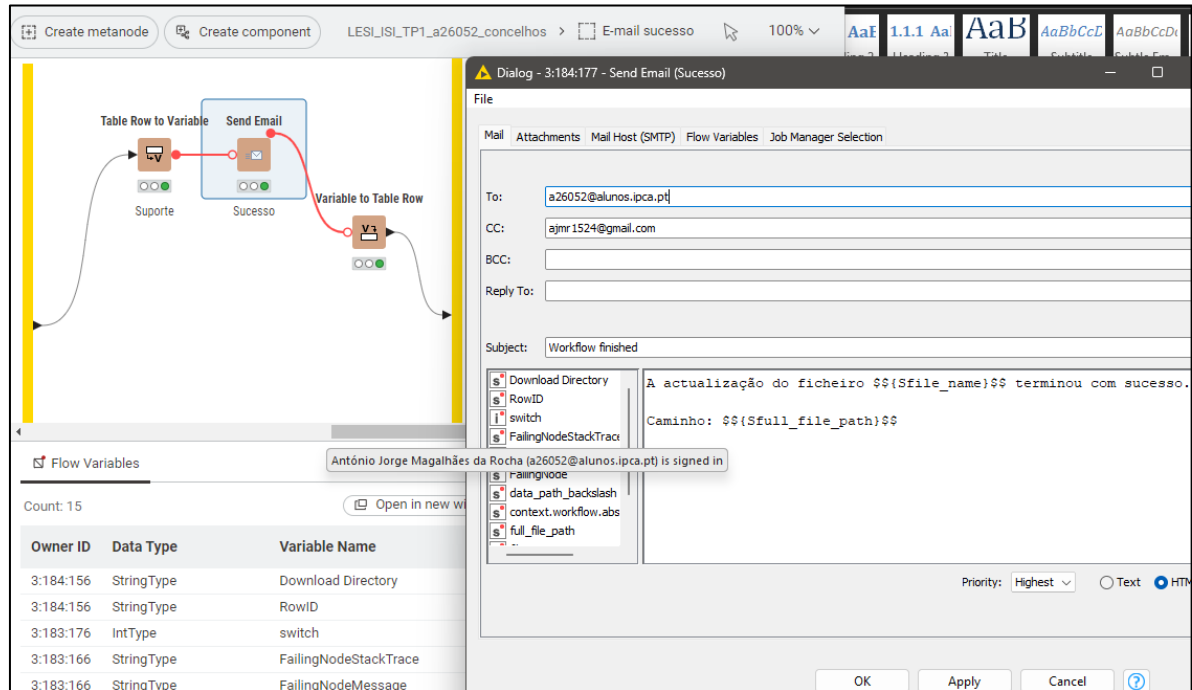
### 5.2.6 Case Switch Start

Conforme o valor da variável “switch”, que lhe é passada pelo “Rule Engine”, executa o caminho da porta 0 ou da porta 1. Neste caso executou a porta zero, como é visível no próprio ícone do node.

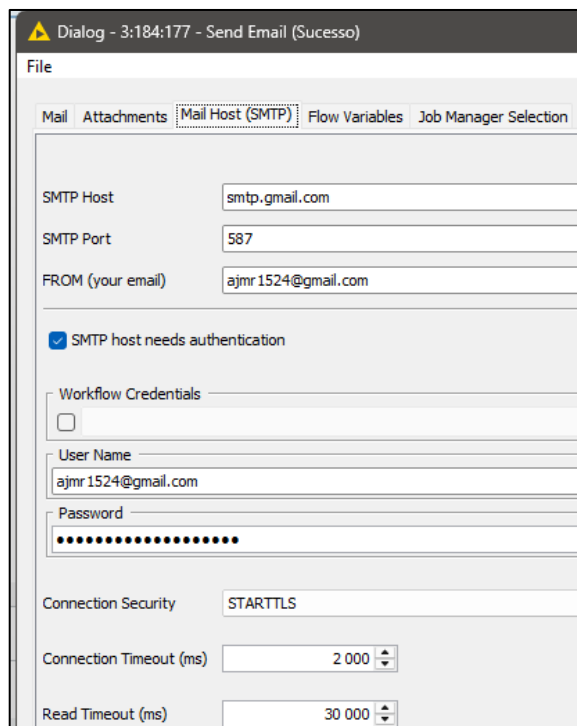


### 5.2.7 E-mail sucesso

Caso o Switch Case execute a porta zero, segue-se o metanode “E-mail sucesso”. Este metanode envia um email de uma conta de e-mail configurada para os destinatários indicados. É possível incluir *placeholders* de variáveis do fluxo no e-mail (ex: `file_name`, `full_file_path`).

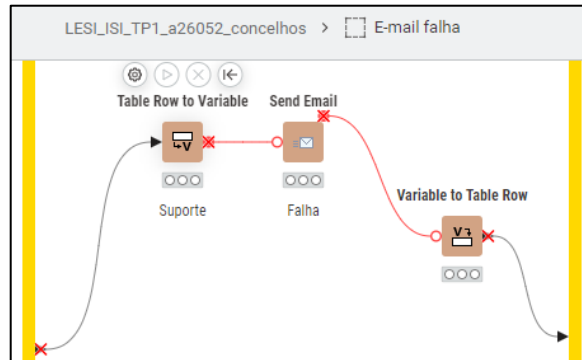


Para que funcione é necessário configurar o Mail Host:



### 5.2.8 E-mail falha

Caso o Switch Case execute a porta 1, é executado o metanode “E-mail falha” e é enviado um e-mail similar da mesma forma, mas de conteúdo diferente.



### 5.2.9 Case Switch End

O caso Switch End sincroniza os caminhos anteriores possíveis. Não é obrigatório e por vezes não faz sentido (ex: quando despoleta apenas visualização de dados).

### 5.2.10 Table Row to Variable e CSV Reader

O node “Table Row to Variable” é necessária para fazer output das variáveis do fluxo num formato que o “CSV Reader” possa utilizar.

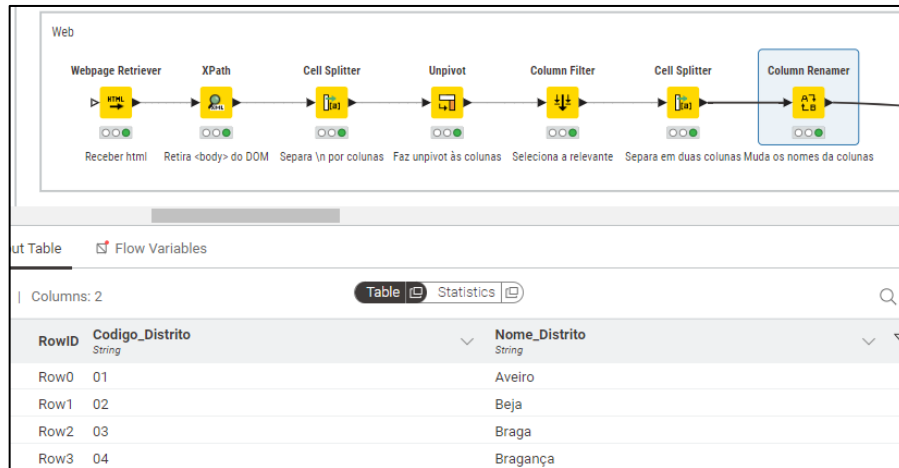
Isto permite configurar o “CSV Reader” para aceder ao CSV com caminho definido na variável “full\_file\_path\_location” e manter o fluxo agnóstico à máquina que o executa.

Isto termina com sucesso o objetivo deste ramo “Concelhos”: obter uma tabela com o código de distrito, código de concelho e nome do concelho.

Row ID	S cod_dis...	S cod_co...	S nome_con...
Row0	01	06	Castelo de Paiva
Row1	01	07	Espinho
Row2	01	08	Estarreja
Row3	01	09	Santa Maria da...
Row4	01	10	Ilhavo

## 5.3 Distritos

O ramo que cria uma tabela com `Codigo_Distrito` e `Nome_Distrito` baseia-se em fazer *retrieve* a uma webpage através do URL e manipular células, listas e colunas.



### 5.3.1 Webpage Retriever

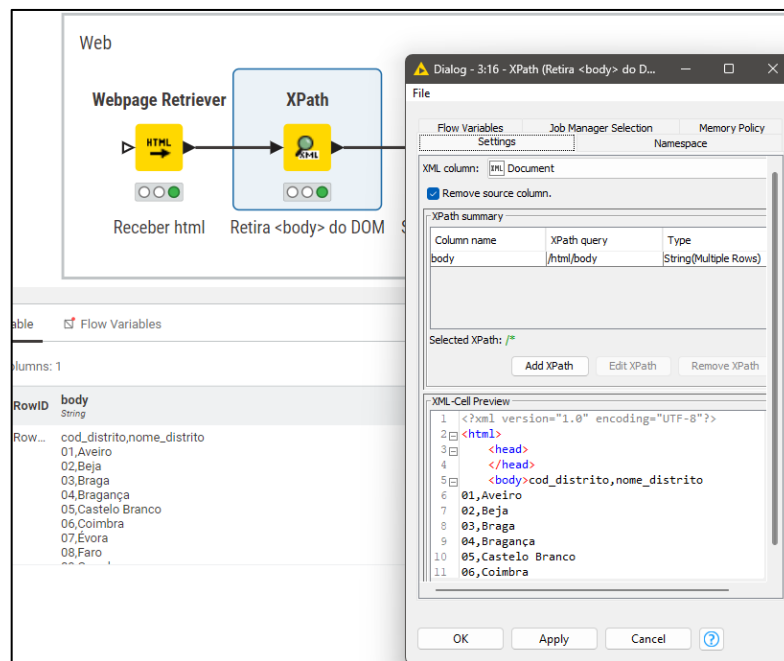
O node “Webpage Retriever” acede à informação raw de uma base de dados no GitHub.



### 5.3.2 XPath

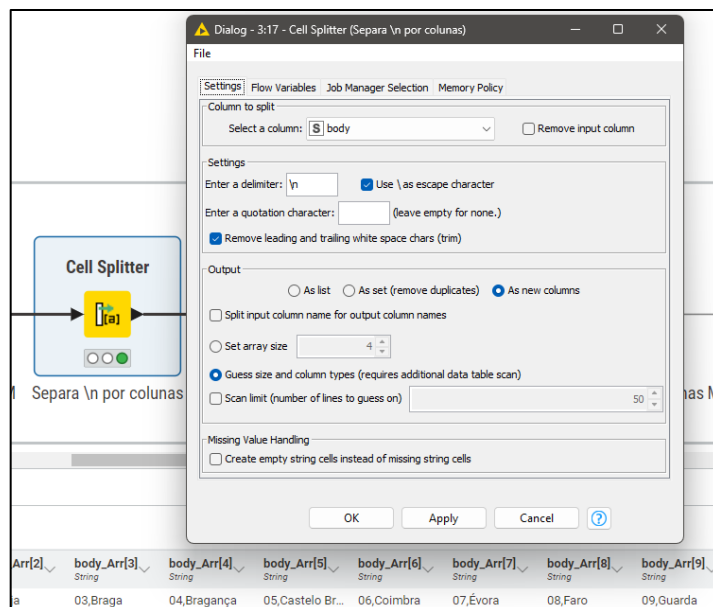
Utilização do *node* “XPATH” para obter o conteúdo do elemento *body* da webpage.

O body contém os código de distrito e o nome dos distritos, separados por vírgulas e mudança de linha.



### 5.3.3 Cell Splitter

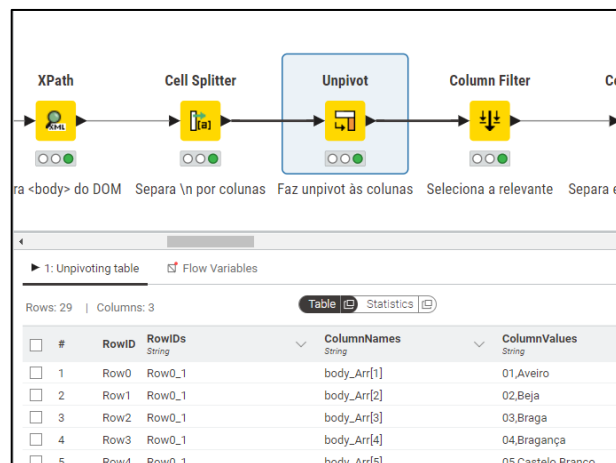
Em sequência do XPATH, é utilizado o Cell Splitter que criar uma coluna para cada lista de pares ordenados [cod\_distrito,nome\_distrito].



### 5.3.4 Unpivot

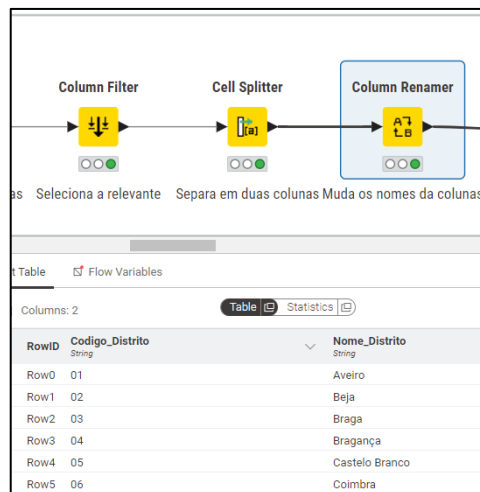
De seguida, o Unpivot passa todas essas colunas para uma coluna em forma de string, e não em lista. Apesar de parecer similar ao conteúdo do body, a grande diferença é que neste ponto cada par corresponde a apenas uma row da tabela.





### 5.3.5 Cell Splitter (2)

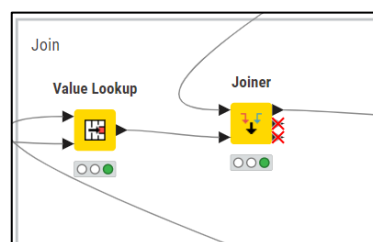
Um novo cell splitter cria duas colunas com base no carácter “,”. Obtemos assim o pretendido: uma tabela como Codigo\_Distrito e Nome\_Distrito.



## 5.4 Cruzar tabelas: Códigos Postais e Ruas, Concelhos, Distritos

Após extrair, guardar, manipular, corrigir e normalizar as três fontes de informação, está na hora de as combinar para gerar uma tabela única de moradas completas, tal como pretendido.

Para isso, serão utilizados dois nodes típicos: “Value Lookup” e “Joiner”.



### 5.4.1 Value Lookup

O node “Value Lookup” é, comparativamente ao “Joiner”, limitado, já que apenas permite acrescentar informações a linhas existentes na tabela original através de uma única coluna “key”. Neste caso, a tabela original, input 1, é a lista completa de moradas e códigos postais. A segunda tabela, input 2, é a tabela que contém a correspondência entre o código de distrito e o nome do distrito.

A correspondência entre tabelas acontece pelas keys “Cod\_Distrito” e “Codigo\_Distrito”. O nome da coluna não tem de ser igual, o que importa é o conteúdo que as mesmas têm em cada uma das suas linhas.

A coluna que é acrescentada à tabela original é a coluna “Nome\_Distrito”, como podemos verificar.

RowID	Cod_Distrito <i>String</i>	Cod_Concelho <i>String</i>	Localidade <i>String</i>	Cod_Postal7 <i>String</i>	Rua <i>String</i>	Nome_Distrito <i>String</i>
Row8	01	01	Agadão	3750-019		Aveiro
Row9	01	01	Aguada De Baixo	3750-996		Aveiro
Row10	01	01	Aguada De Baixo	3750-031		Aveiro
Row11	01	01	Aguada De Baixo	3750-033		Aveiro
Row12	01	01	Aguada De Baixo	3750-035		Aveiro
Row13	01	01	Aguada De Cima	3750-043	Rua das Almas	Aveiro
Row14	01	01	Aguada De Cima	3750-041	Rua do Sabugueiro	Aveiro
Row15	01	01	Aguada De Cima	3750-041	Rua da Azenha	Aveiro
Row16	01	01	Aguada De Cima	3750-043	Vila das Quintas	Aveiro

### 5.4.2 Joiner

O Joiner permite configurações bastante mais poderosas e complexas do que o “Value Lookup”.

No exemplo, o primeiro input é a tabela de Concelhos e o segundo input é a tabela resultante do “Value Lookup” anterior. O objectivo é acrescentar à segunda tabela o nome do Concelho. Para atingir este objetivo é necessário estabelecer uma correspondência de duas keys entre as tabelas, o Código de Distrito e o Código de Concelho. Isto deve-se ao facto de o Código de Concelho não

identificar inequivocamente um Concelho, apenas a combinação de Código de Distrito e Código de Concelho formam uma *unique key*.

Para controlo de erro e qualidade da informação final, são seleccionadas apenas as linhas que contêm nome do Concelho, partindo do princípio que qualquer morada válida deverá ter um nome do Concelho válido.

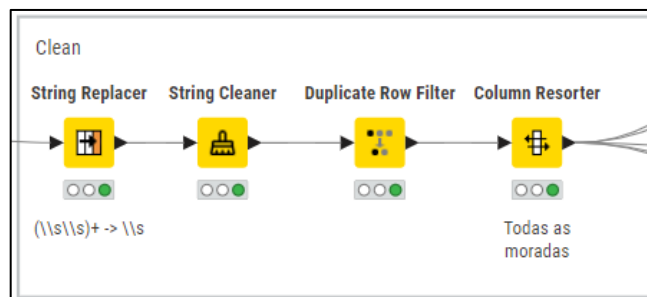
The screenshot shows a 'Dialog - 3:278 - Joiner' window. Under 'Matching Criteria', the 'Match' section has 'All of the following' selected. Two criteria are defined: 'Top input (left table)' with 'cod\_distrito' and 'Bottom input (right table)' with 'Cod\_Distrito'; and 'Top input (left table)' with 'cod\_concelho' and 'Bottom input (right table)' with 'Cod\_Concelho'. The 'Compare values in join columns by' dropdown is set to 'Value and type'. In the 'Include in Output' section, 'Matching rows' and 'Right unmatched rows' are checked, accompanied by a Venn diagram. The 'Output Columns' section shows 'Manual' selected, with 'Excludes' containing 'cod\_distrito' and 'cod\_concelho', and 'Includes' containing 'nome\_concelho'.

Isto concluí o objetivo de cruzar as três tabelas de informação, produzindo uma tabela de moradas com nomes de concelho e distritos e com algumas moradas inválidas já filtradas.

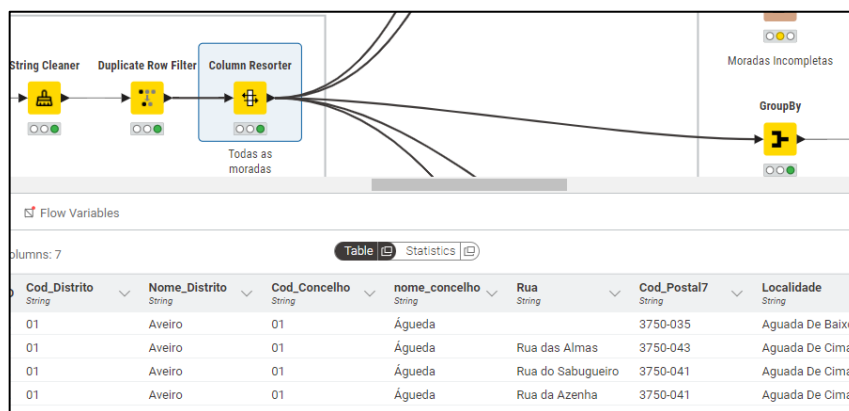
RowID	nome_concelho <small>String</small>	Cod_Distrito <small>String</small>	Cod_Concelho <small>String</small>	Localidade <small>String</small>	Cod_Postal7 <small>String</small>	Rua <small>String</small>	Nome_Distrito <small>String</small>
Row...	Águeda	01	01	Agadão	3750-019		Aveiro
Row...	Águeda	01	01	Aguada De Baixo	3750-996		Aveiro
Row...	Águeda	01	01	Aguada De Baixo	3750-031		Aveiro
Row...	Águeda	01	01	Aguada De Baixo	3750-033		Aveiro
Row...	Águeda	01	01	Aguada De Baixo	3750-035		Aveiro
Row...	Águeda	01	01	Aguada De Cima	3750-043	Rua das Almas	Aveiro
Row...	Águeda	01	01	Aguada De Cima	3750-041	Rua do Sabugueiro	Aveiro
Row...	Águeda	01	01	Aguada De Cima	3750-041	Rua da Azenha	Aveiro

## 5.5 Limpar

O passo seguinte é limpar a tabela, tanto a nível de colunas como a nível de células e linhas.



Para efetuar a limpeza, é utilizado um “String Replacer” para substituir duplos espaços para apenas um espaço. De seguida, um “String Cleaner” limpa espaços no início e no final das *strings*. É aplicado um filtro para eliminar eventuais linhas inteiramente duplicadas e uma ordenação das colunas numa ordem mais perceptível.

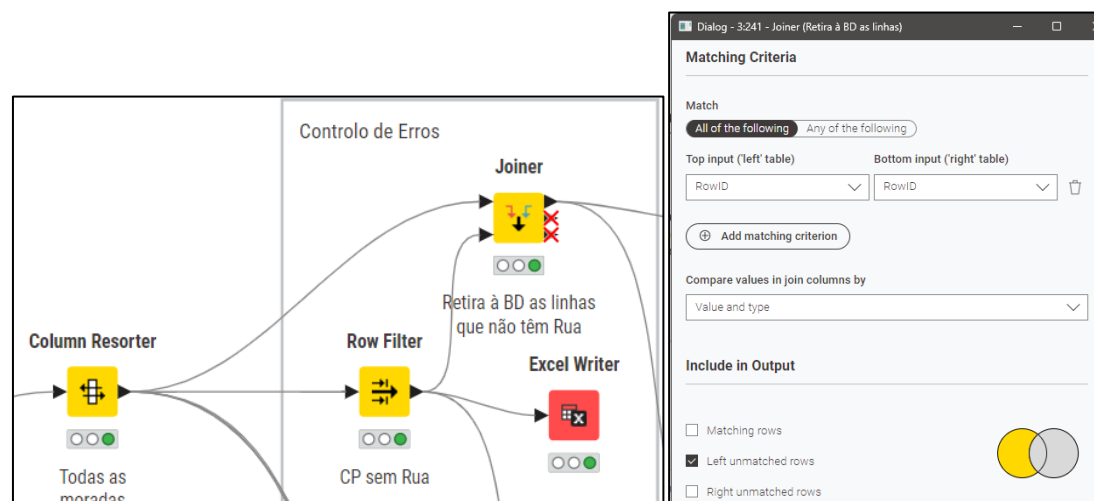


## 5.6 Controlo de erros

No passo seguinte são identificadas, através de um filtro, as moradas que não contêm uma Rua válida.

Estas moradas são então guardadas num ficheiro Excel para que possam ser identificadas e analisadas.

É então utilizado um Joiner para retirar (através de um Left Anti) à tabela original as rows da tabela de moradas inválidas.



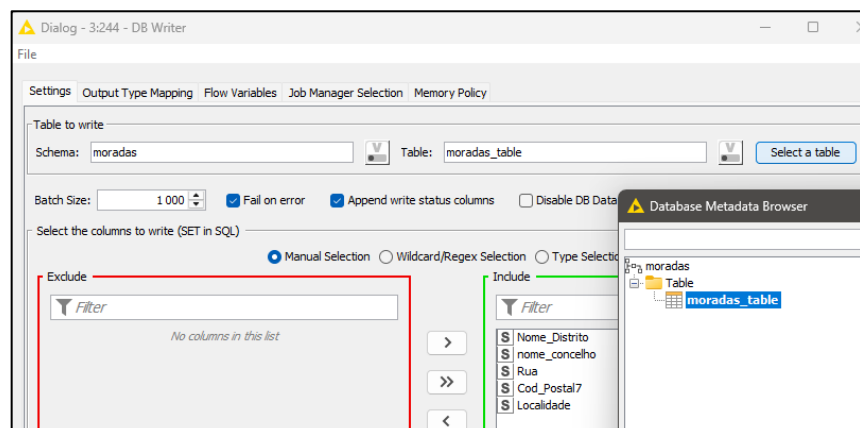
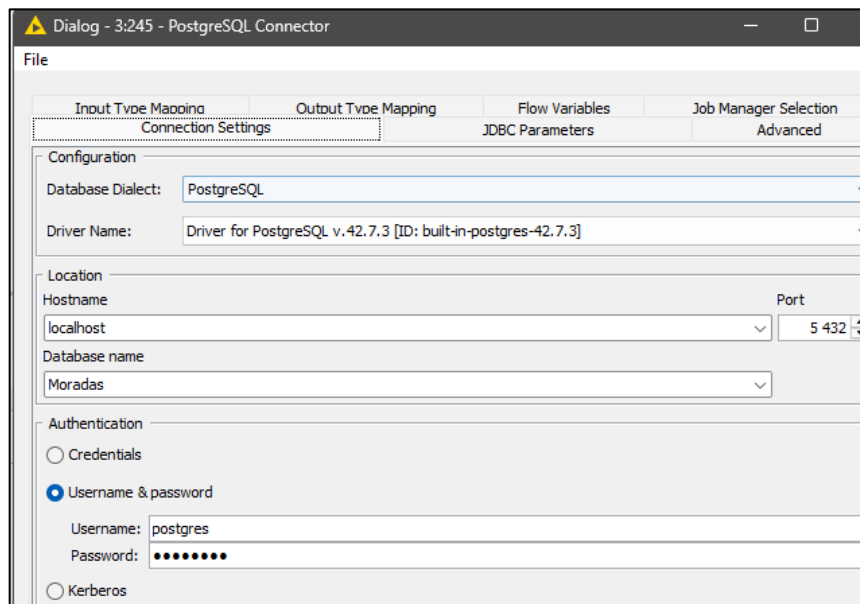
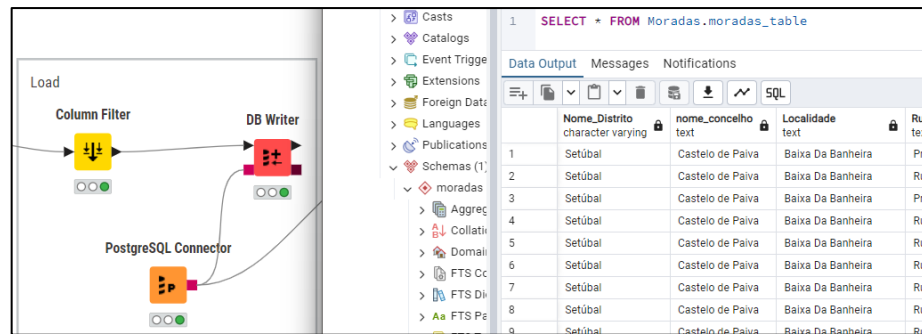
Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).

## 5.7 Load

### 5.7.1 Escrever na Base de Dados

De seguida temos a fase de “Load”, ou carregamento dos dados.

Para mais informações sobre conectores de bases de dados ler o capítulo 9.1 .

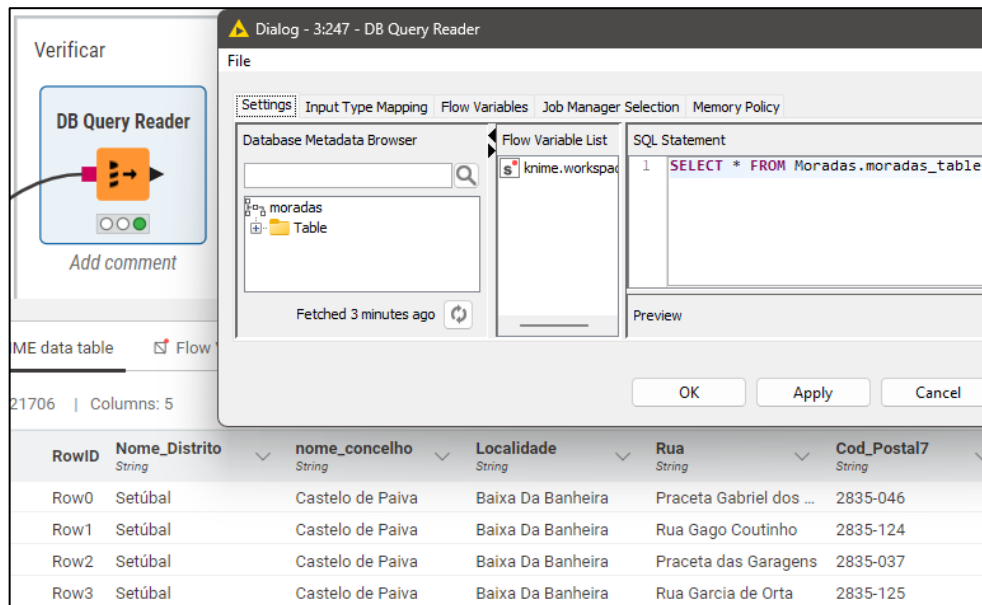


Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).

### 5.7.2 Query à Base de dados

Como podemos verificar pela query “SELECT \* FROM Moradas.moradas\_table” na imagem anterior, os dados finais são carregados com sucesso para uma base dados em PostgreSQL.

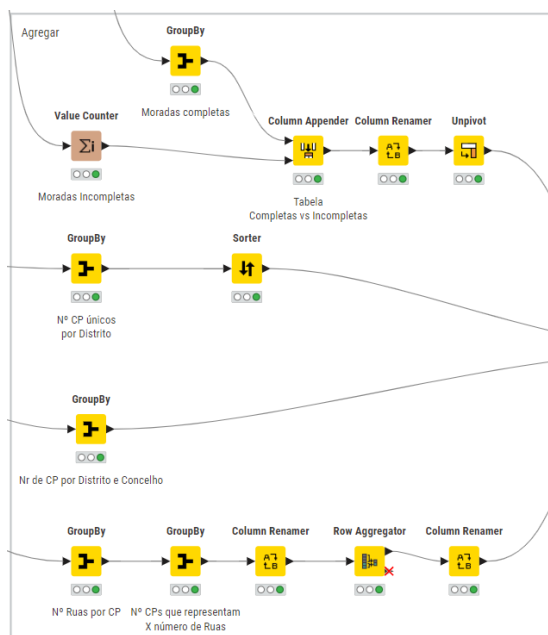
No entanto, foi também configurado um node Knime para executar essa query.



## 5.8 Agregar, analisar e visualizar

Embora o objetivo de ter um base de dados único com todas as moradas completas válidas já esteja concluído no passo anterior, foram realizadas ações de agregação, análise e visualização de dados.

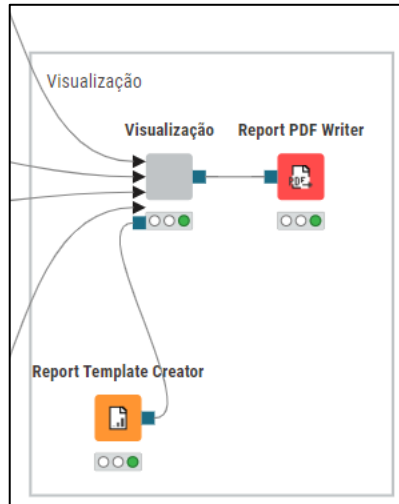
Na fase de Agregar são realizados vários somatórios e médias por categoria/grupo que pretendem representar e avaliar a qualidade do resultado.



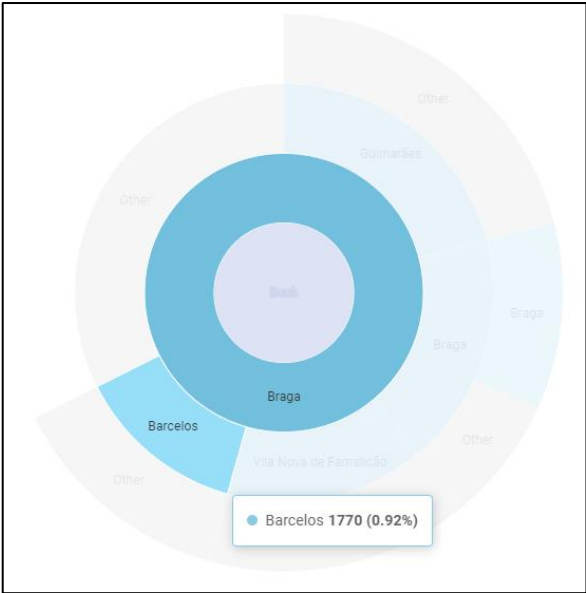
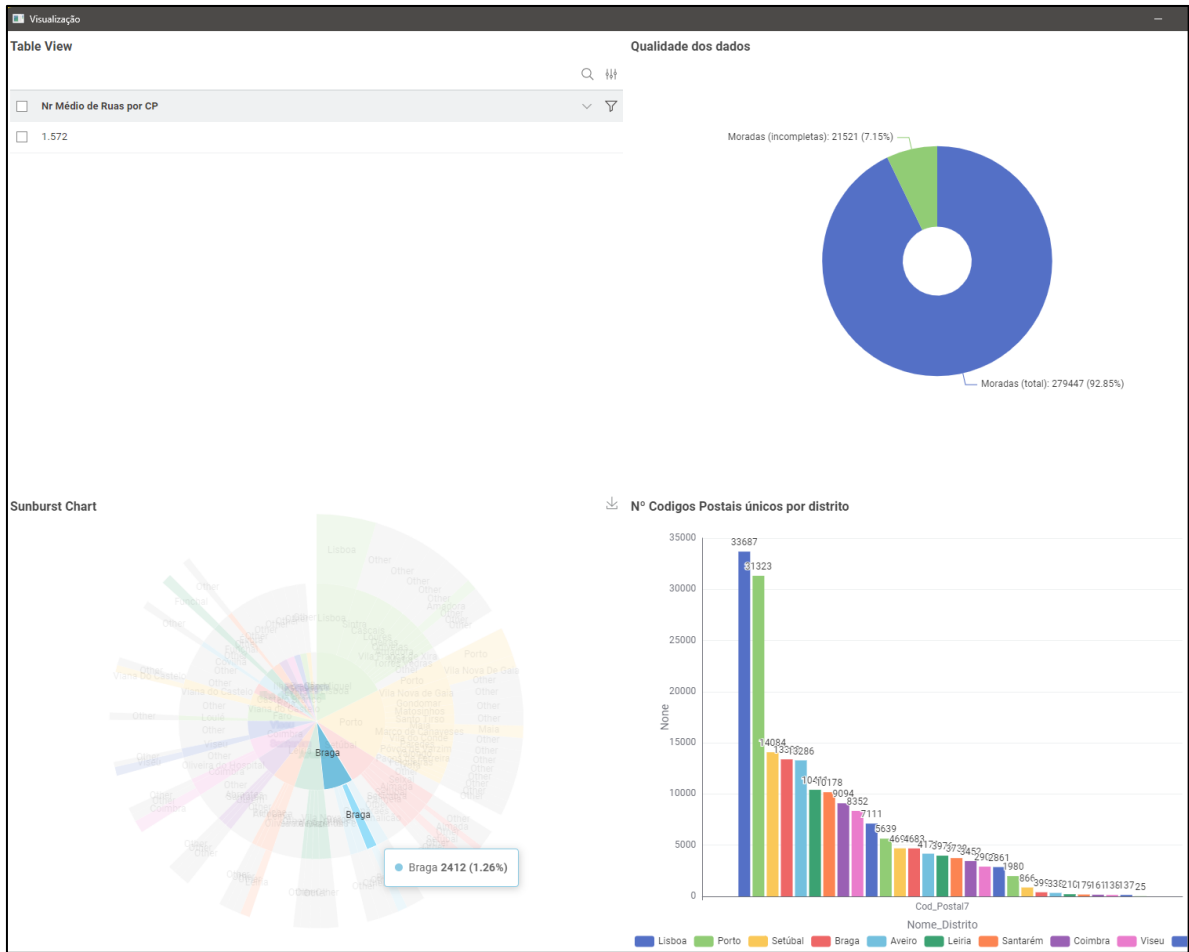
Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).

Estas manipulações geram gráficos que serão incluídos numa dashboard e num relatório pdf.

No entanto, para que seja possível incluí-los numa dashboard interativa, os mesmos devem ser agregados num “Component”, designado neste caso por “Visualização”.



A *dashboard* mostra o número média de ruas por Código-Postal de 7 dígitos, a percentagem de moradas incompletas (7,5%) que foram excluídas da base de dados final, o número de códigos postais únicos em cada distrito e um gráfico “Sunburst” onde podemos navegar por distrito, concelho e localidade e verificar o número e percentagem de códigos postais que compõem a base de dados.





## 6 WORKFLOW: API Condições Rodoviárias – alerta e-mail

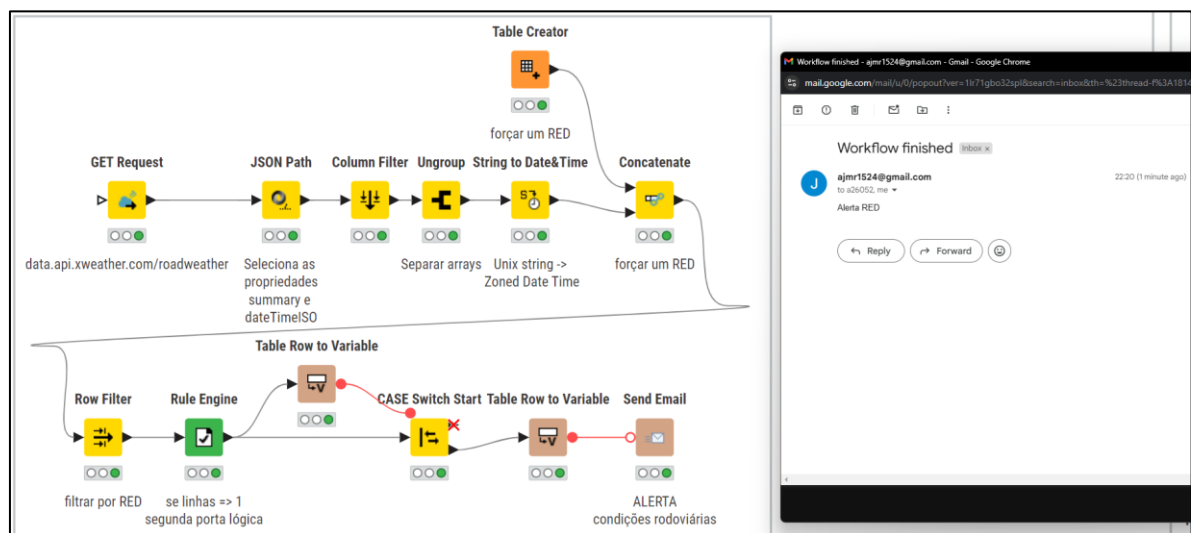
No exemplo abaixo, podemos verificar um fluxo que faz GET ao endpoint “roadweather” da API da xweather” e retorna um JSON que contém informações sobre as condições rodoviárias.

Nesse JSON, seleciona apenas os atributos dateTimeISO e summary, que contêm, em duas listas, a data e hora, assim como o estado das vias (GREEN, YELLOW e RED).

Depois de separar as listas em variáveis e filtrar apenas por linhas com summary = “RED”, um “Rule Engine” determina a porta lógica a executar no Case Switch mais à frente.

Se existir pelo menos uma linha com “RED”, o Rule Engine atribui à variável “Fim\_do\_Mundo” o valor 1, que definirá a execução da porta 1 do Case Switch (em detrimento da porta 0).

A porta 1 é responsável por despoletar o envio de um e-mail prioritário, como é possível verificar na seguinte imagem.



## 7 WORKFLOW EXTRA A: Grupos Etários .table

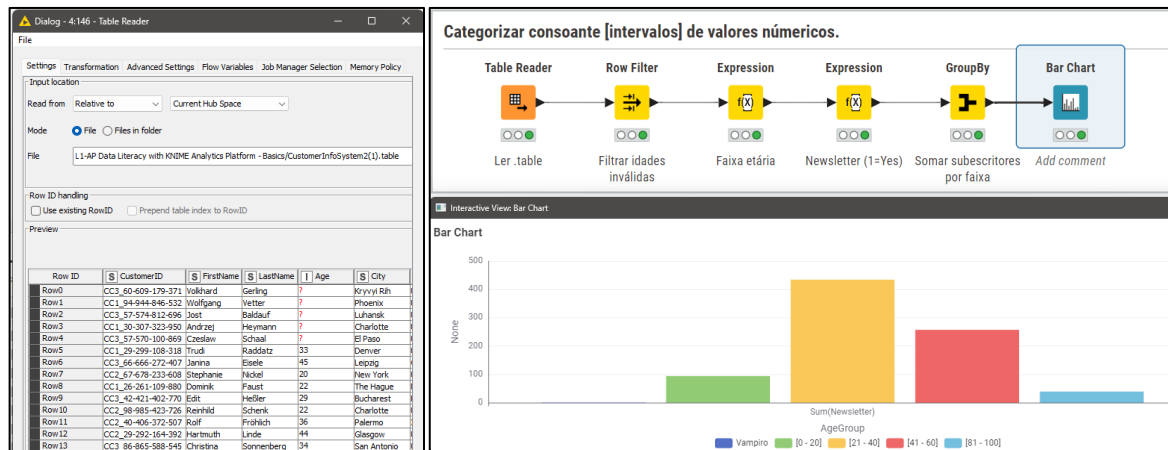
A obtenção de dados em ficheiros `.table`, utilizados pelo Knime, é extremamente simples. Basta utilizar um node “Table Reader” apontado para o ficheiro. Neste exemplo, são obtidos os dados de subscritores de uma newsletter.

Após obter uma lista de subscritores, são filtradas as linhas com “Age” inválida. É então criada uma coluna com a faixa etária com base na idade, utilizando a seguinte expressão:

Expression editor

```
1 if(S["Age"] <= 20, "[0 - 20]", S["Age"] <= 40, "[21 - 40]", S["Age"] <= 60, "[41 - 60]",
2 S["Age"] <= 60, "[61 - 80]", S["Age"] <= 100, "[81 - 100]", S["Age"] <= 120, "[101 - 120]", "Vampiro")
```

Por último, é feito um somatório do número de subscritores em cada faixa etária e essa informação é disponibilizada num gráfico de barras.



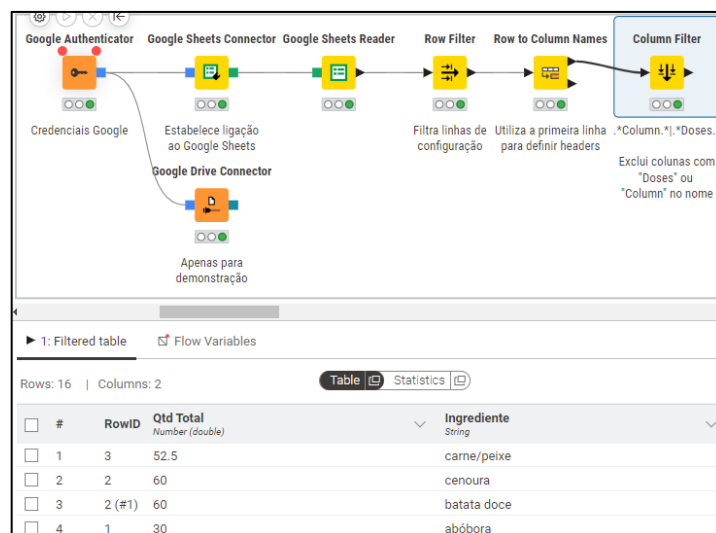
## 8 WORKFLOW EXTRA B: Receita Sopa – Google Drive e Regex

Na figura seguinte podemos verificar um exemplo de obtenção de dados de uma receita de sopa através de um node de autenticação da Google, de um conector ao Google Sheets e da posterior seleção e leitura de uma folha de um livro de cálculo.

Após obtenção do livro, são filtradas linhas que não pertencem à receita.

A primeira linha resultante é, na verdade, o header da coluna e é utilizado para tal.

Por último, é utilizada a expressão regular `<.*Column.*|.*Doses.*>` para filtrar todas as colunas com “Column” (coluna vazias) ou “Doses” no nome, que são as colunas que não pretendemos manter.



O KNIME mostra dinamicamente a aplicação da expressão regular no menu de configuração:



## 9 Casos de uso / experiências / considerações

### 9.1 Base de dados SQL

#### 9.1.1 Conexão: PostgreSQL e MySQL

A conexão a bases de dados em PostgreSQL é bastante simples, sendo apenas necessário um *node* conector para conector o Knime à base de dados e outro *node* para selecionar, dentro da mesma, a tabela desejada.

A imagem seguinte demonstra o *output* de uma *query* SQL à base de dados e o *output* do *node* “DB Table Selector”, que contêm os mesmos dados e o mesmo tipo de variáveis (ou equivalente, no caso de VARCHAR).

**Knime Workflow Diagram:**

SQL Databases

PostgreSQL Connector → DB Table Selector

1: DB Data | Flow Variables

#	RowID	id	name
		Number (integer)	String
1	Row0	1	um
2	Row1	2	dois

**pgAdmin 4 Query Result:**

```
SELECT * FROM employees;
```

id	name
[PK] integer	character varying (100)
1	um
2	dois

A conexão ao MySQL é idêntica ao PostgreSQL e, na verdade, a qualquer base de dados. Existe até um conector genérico “DB Connector”.



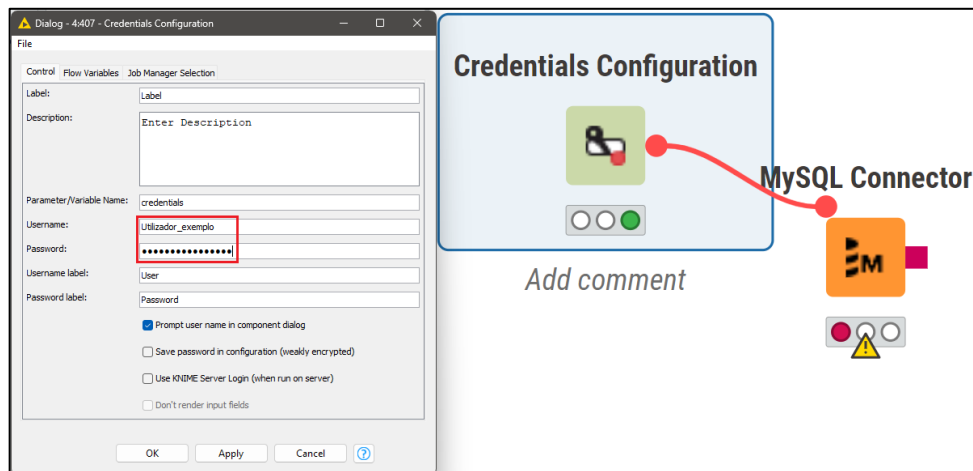
### 9.1.2 Autenticação e segurança

De salientar que, ao contrário do que é mostrado no exemplo do PostgreSQL, é mais correto armazenar as credenciais num node específico para esse propósito, aumentando a segurança do *workflow* e da sua partilha.

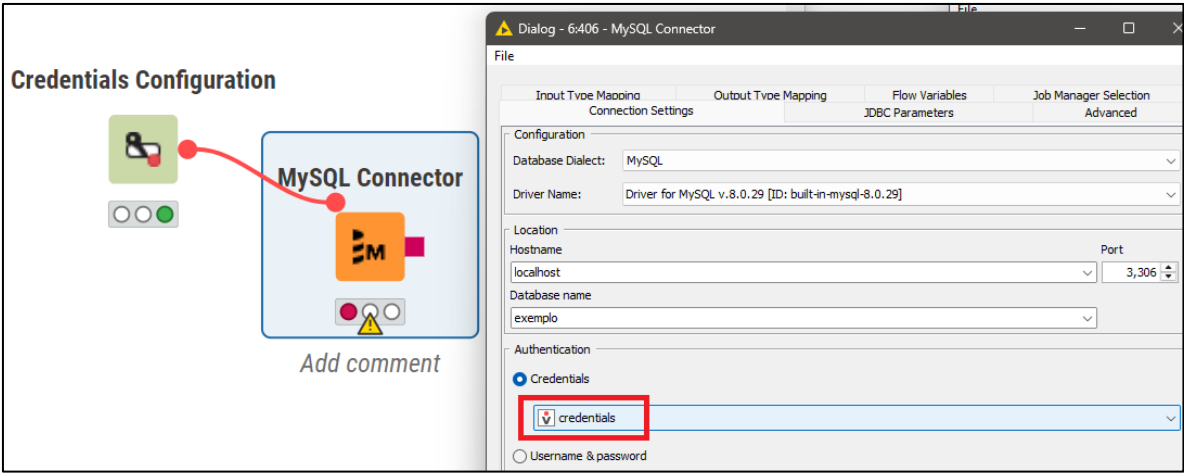
Armazenar as credenciais neste node específico proporciona acesso a vários métodos de requisição, armazenamento e encriptação das credenciais.

Este node de credenciais passa, por variável, as credenciais a serem utilizados no node de conexão.

Na imagem seguinte podemos ver os parâmetros do node de credenciais. É criada a variável “credentials” que irá passar os dados preenchidos neste node ao node conector.

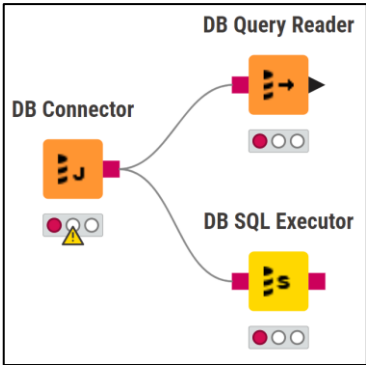


Para utilizar, no conector, as credenciais passadas por variável basta, no campo de autenticação, selecionar a variável “credentials”.



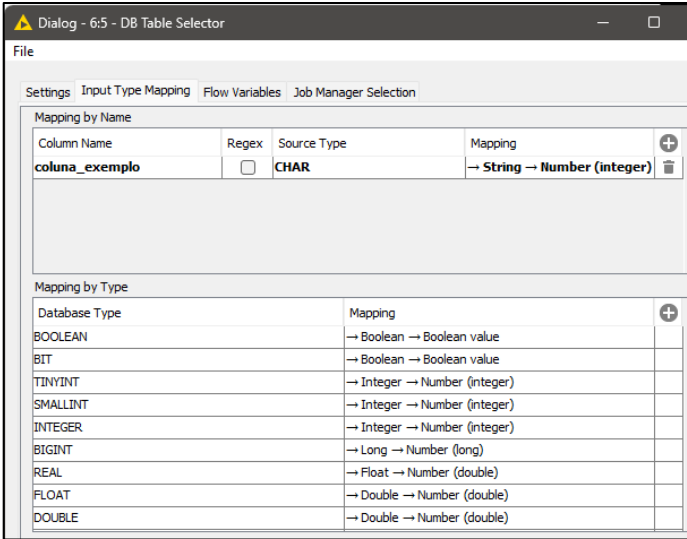
### 9.1.3 Execução de queries

Existem também nodes dedicados para executar e ler o output de queries a bases de dados SQL.



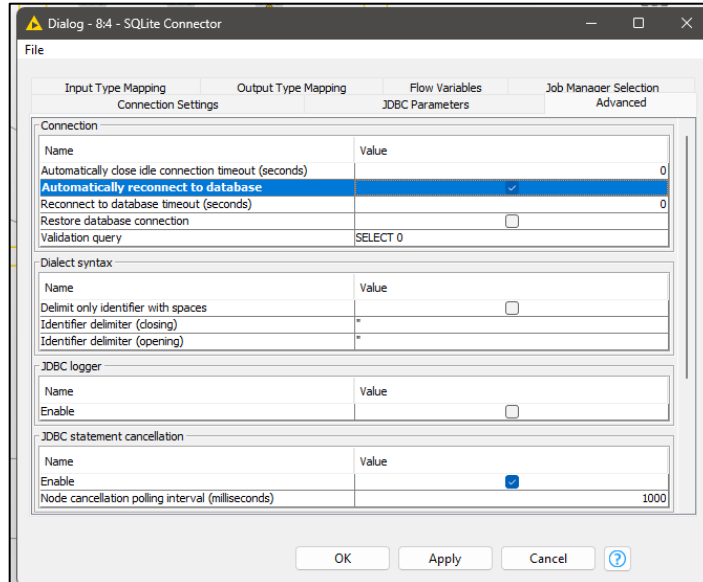
### 9.1.4 Mapeamento de tipos de dados

É também possível definir conversão do tipo de dados de acordo com o tipo de dados da fonte (*Mapping by Type*) ou especificar a conversão para determinadas colunas (*Mapping by Name*):



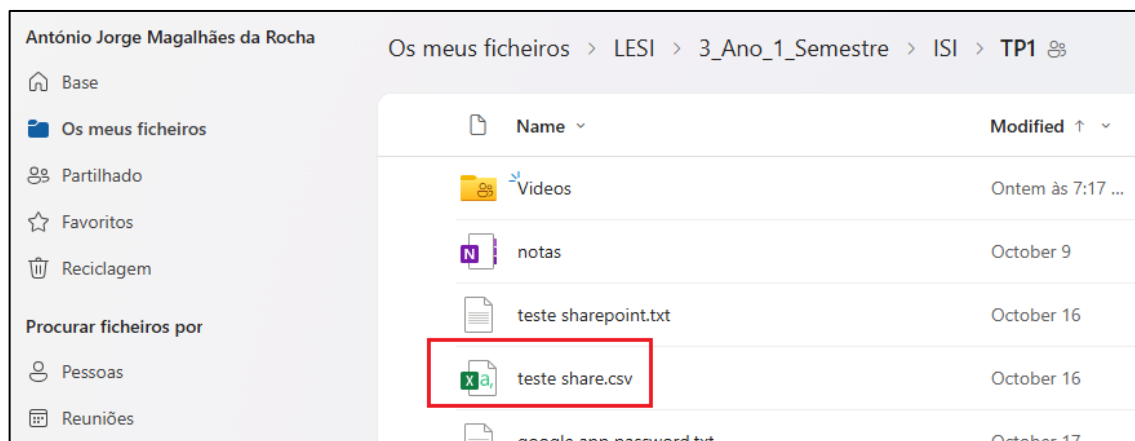
### 9.1.5 Reconexão automática a bases de dados

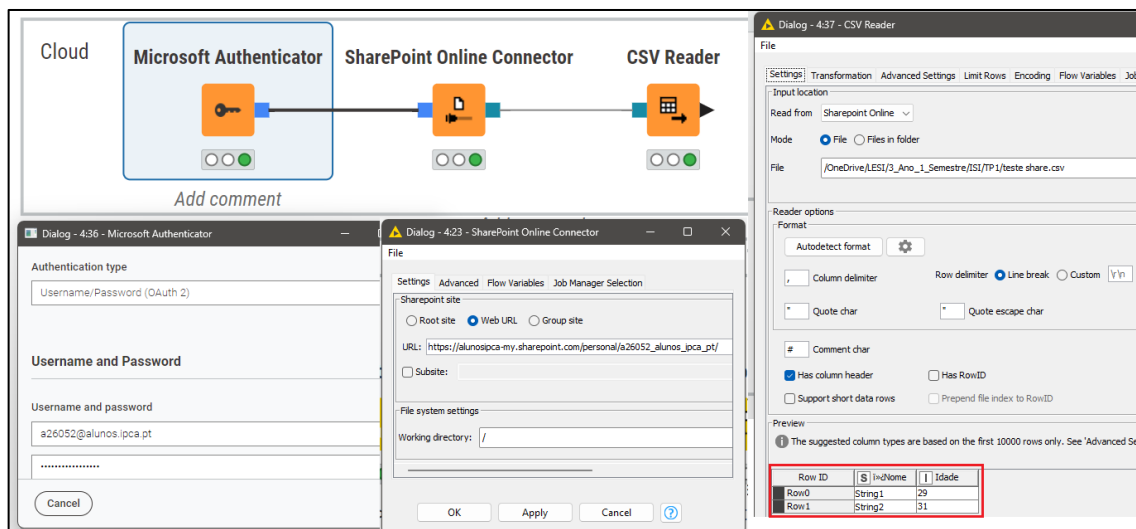
A conexão à base de dados será quebrada sempre que o *workflow* é fechado. Para evitar este comportamento, é possível definir reconexão automática à BD:



## 9.2 Sharepoint

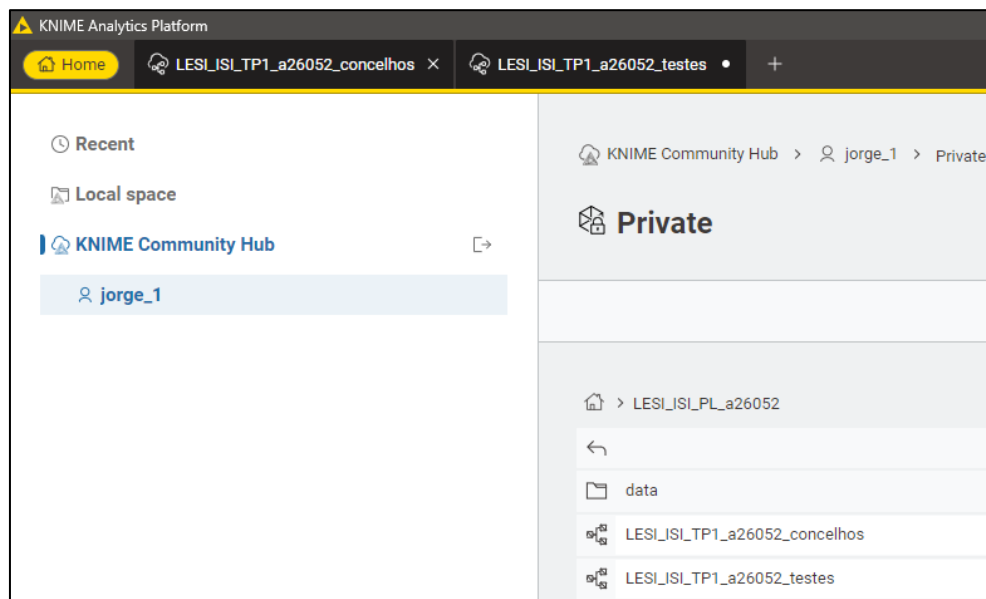
No exemplo seguinte podemos visualizar a autenticação, conexão e dados resultantes de extração de um CSV alojado no Sharepoint.





## 9.3 Knime Community Hub

Os *workflows* encontram-se alojados no servidor do Knime Community Hub, que proporciona armazenamento grátis de *workflows* até 50MB e sincronização baseada na conta de utilizador.



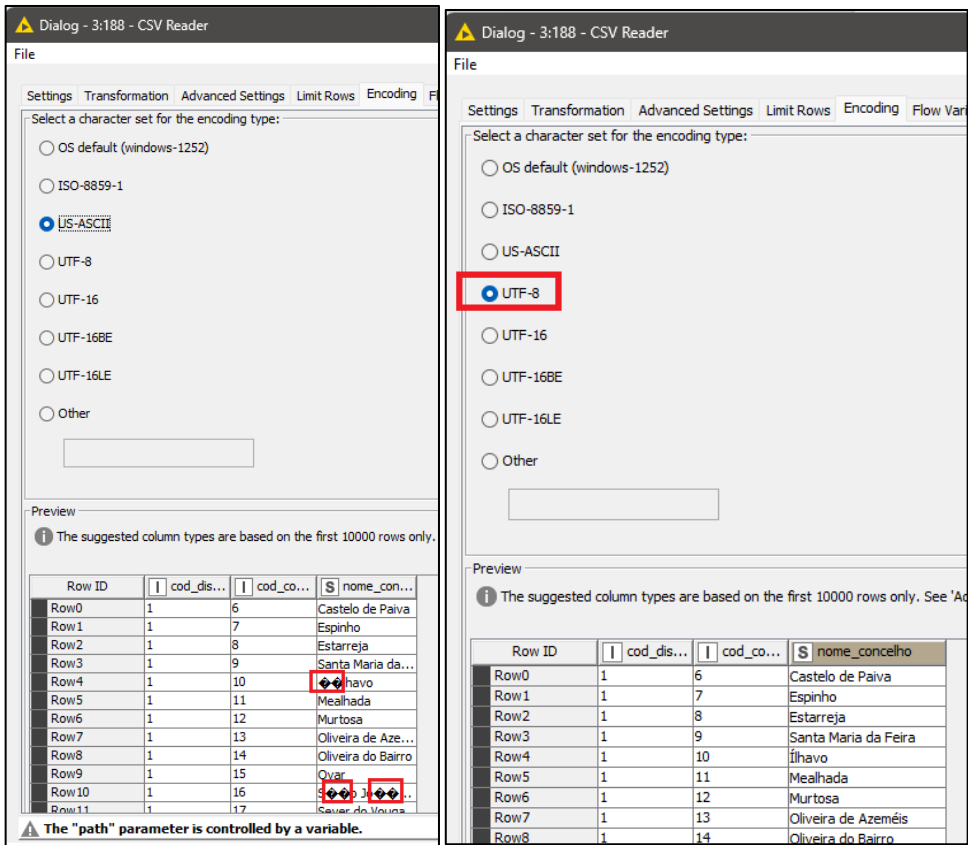
O funcionamento de *workflows* alojados no Knime Community Hub é amplamente suportado graças, nomeadamente, às hipóteses de caminhos relativos e aos nodes de “contexto” (ver capítulo 7).

## 9.4 Codificação de caracteres

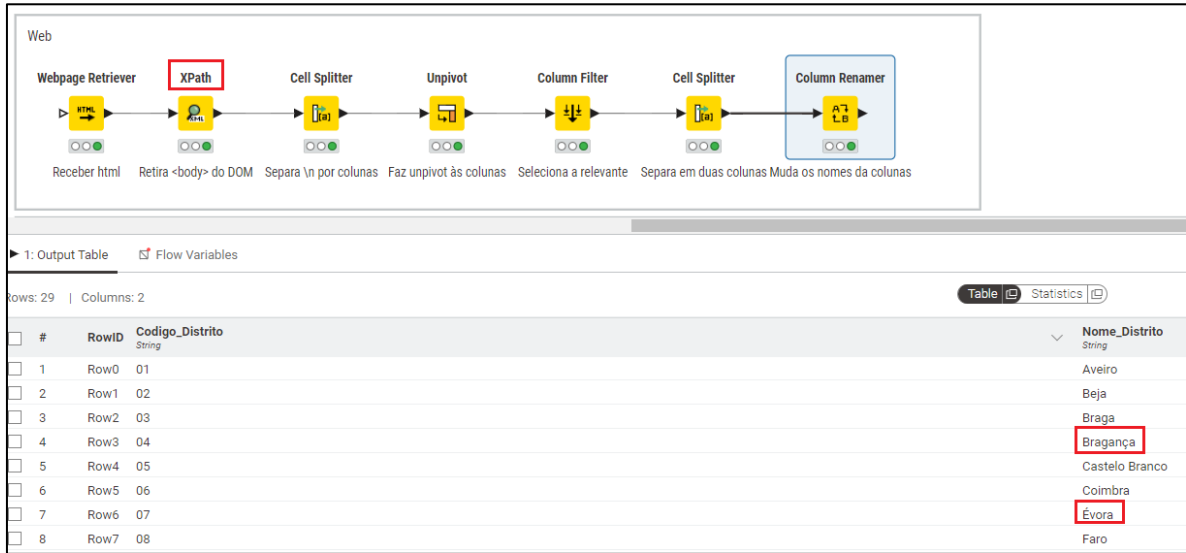
### 9.4.1 Caracteres especiais

Foi necessário definir a codificação de texto como UTF-8, e não o *default*. Caso contrário os caracteres especiais não serão corretamente representados:

Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).



No caso do acesso por XPATH não é necessária esta configuração ao nível da leitura de texto. Já funciona por *default*:



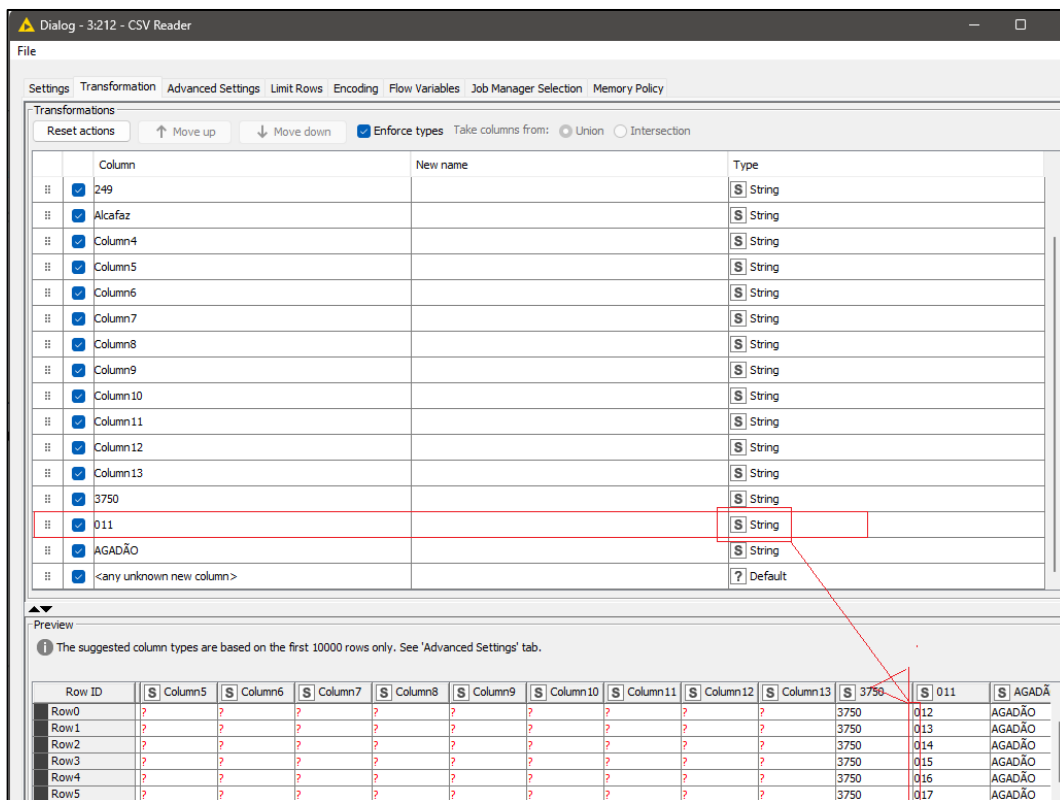
### 9.4.2 Leading Zeros

Foi necessário alterar os tipos de variáveis detetados automaticamente durante o *import* de CSV.

Caso contrário, não apresentaria os *leading zeros* dos códigos postais.

Desenvolvimento de processos e aplicação de ferramentas de ETL (Extract, Transform, Load).





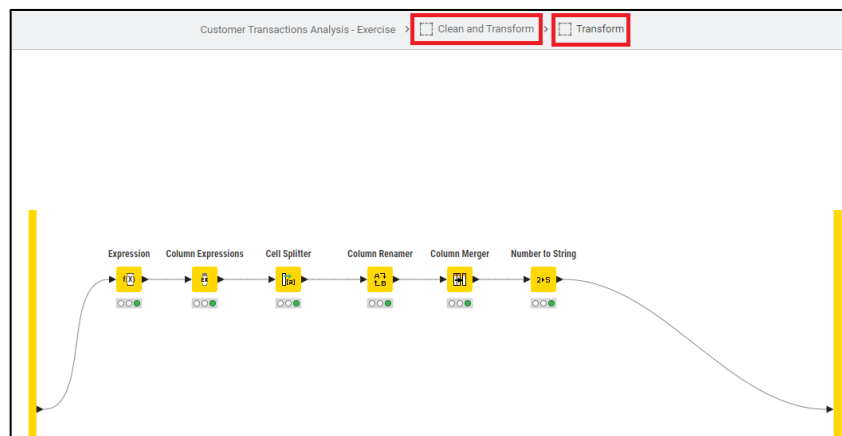
## 9.5 Abstração, representação e documentação gráfica do processo

O requisito de auxiliar na documentação do processo e na facilidade de abstração e encapsulamento através dos “Components”, que são essencialmente novos nodes criados a partir de outros e “Metanodes”, que servem apenas para agrupar visualmente conjuntos de nodes que têm uma finalidade específica e divisível do resto do workflow.

## 9.6 Metanodes de Metanodes

É possível ter um *metanode* dentro de outro *metanode*.

Neste caso existe um *metanode* chamado “Clean and Transform” que, por sua vez, contem um *metanode* chamado “Transform”.



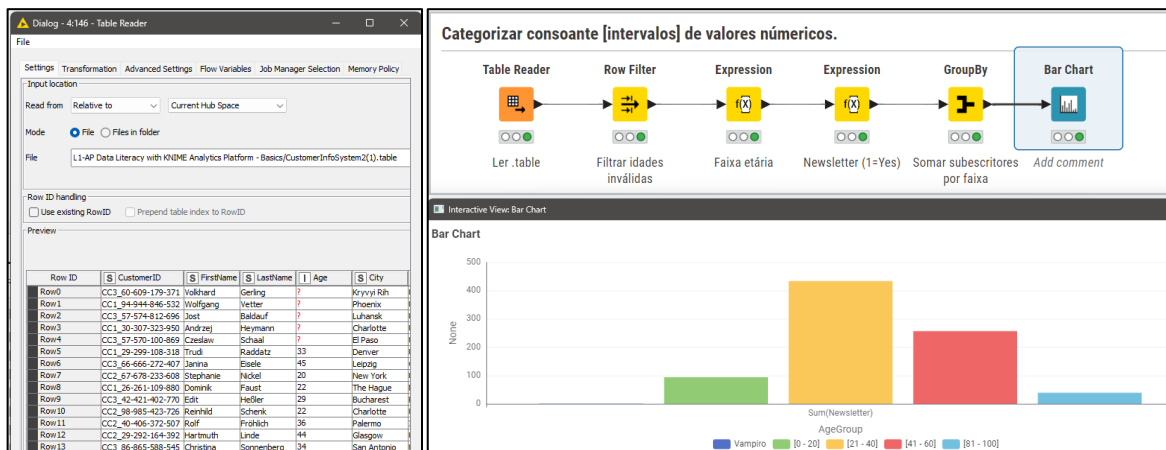
## 9.7 GroupBy : Categorização e agregação

Após obter uma lista de subscritores, são filtradas as linhas com “Age” inválida. É então criada uma coluna com a faixa etária com base na idade, utilizando a seguinte expressão:

Expression editor

```
1 if([${"Age"}] <= 20, "[0 - 20]", ${"Age"} <= 40, "[21 - 40]", ${"Age"} <= 60, "[41 - 60]",
2 ${"Age"} <= 60, "[61 - 80]", ${"Age"} <= 100, "[81 - 100]", ${"Age"} <= 120, "[101 - 120]", "Vampiro"]
```

Por último, é feito um somatório do número de subscritores em cada faixa etária e essa informação é disponibilizada num gráfico de barras.



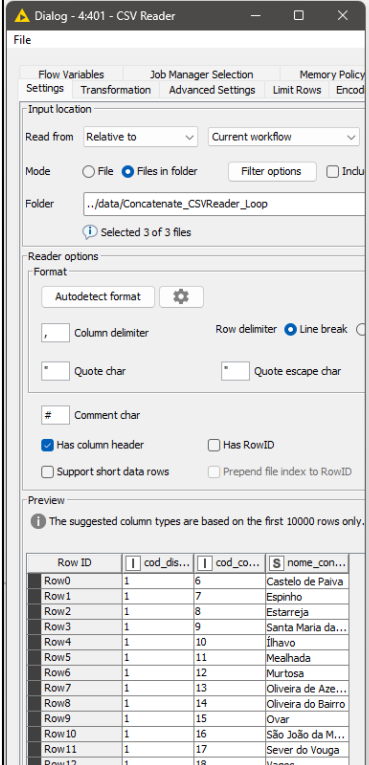
## 9.8 Ler e/ou juntar ficheiros CSV

O Knime permite, pelo menos, três formas de ler e fazer *append* a vários ficheiros .csv.

O mais simples de todos, cuja configuração é visível na imagem, é utilizar o node “CSV Reader” apontado para uma pasta que tenha mais do que um CSV. Ele irá automaticamente fazer *append* aos dados cujos nomes das folhas de cálculo e das colunas sejam idênticos.

É também possível fazer o *append* manual, mas tem a desvantagem de ser necessário saber de antemão quantos e quais são os ficheiros a sintetizar. Poderá ser útil quando se pretende unir ficheiros que se encontram em diretórios diferentes.

Outra forma, que tira proveito da capacidade do Knime de listar diretórios e ficheiros, é utilizar o “*List Files/Folders*” para gerar uma lista de ficheiros e iniciar um ciclo que, para cada elemento da lista gerada, faz a leitura do conteúdo e *append* automático.




### Juntar (append) ficheiros Excel com a mesma estrutura:

Três formas possíveis com o mesmo resultado

#### CSV Reader direcionado para uma pasta a filtrar por .csv

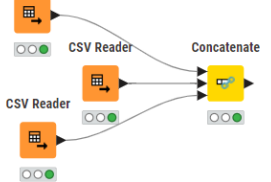
- Lê ficheiros csv dentro de uma pasta específica.
- Faz *append* das tabelas obtidas a partir da leitura dos csv



*Add comment*

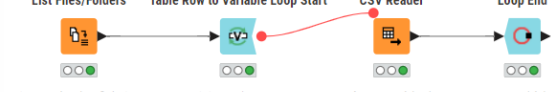
#### Append

- Lê ficheiros csv.
- Faz *append* as tabelas obtidas dos ficheiros csv.



#### Loop Concatenate

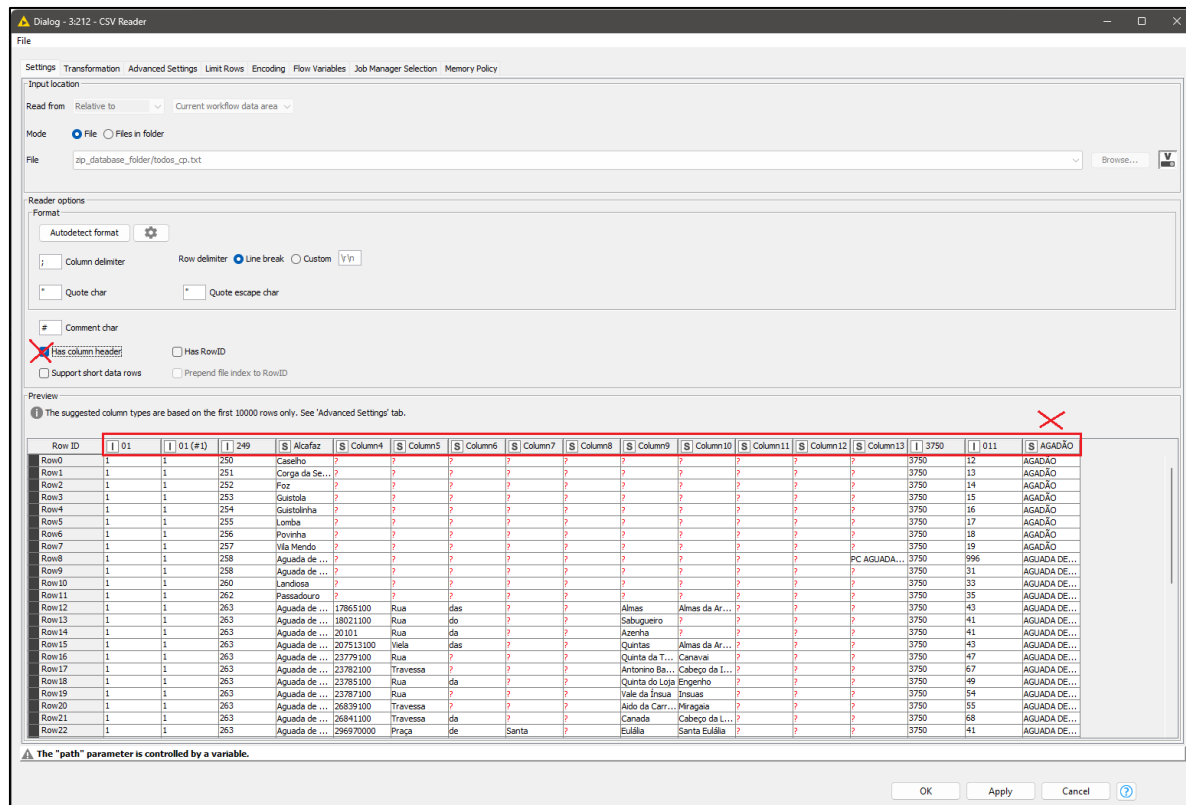
- Lê ficheiros .csv numa pasta.
- Faz loop por todos, fazendo *append* ao seu conteúdo numa única tabela.



Lista paths dos ficheiros .csv dentro de uma pasta    Inicia um loop que percorre todos os paths    Lê o conteúdo do ficheiro    Faz *append* à leitura de todos os ficheiros

## 9.9 Identificação de colunas

Como a fonte de dados não contém nomes de colunas, foi também necessário desligar a opção de os ler. Caso contrário, o Knime assumia a primeira linha (uma morada) como nomes das colunas.



## 10 Vídeo com demonstração (QR Code)

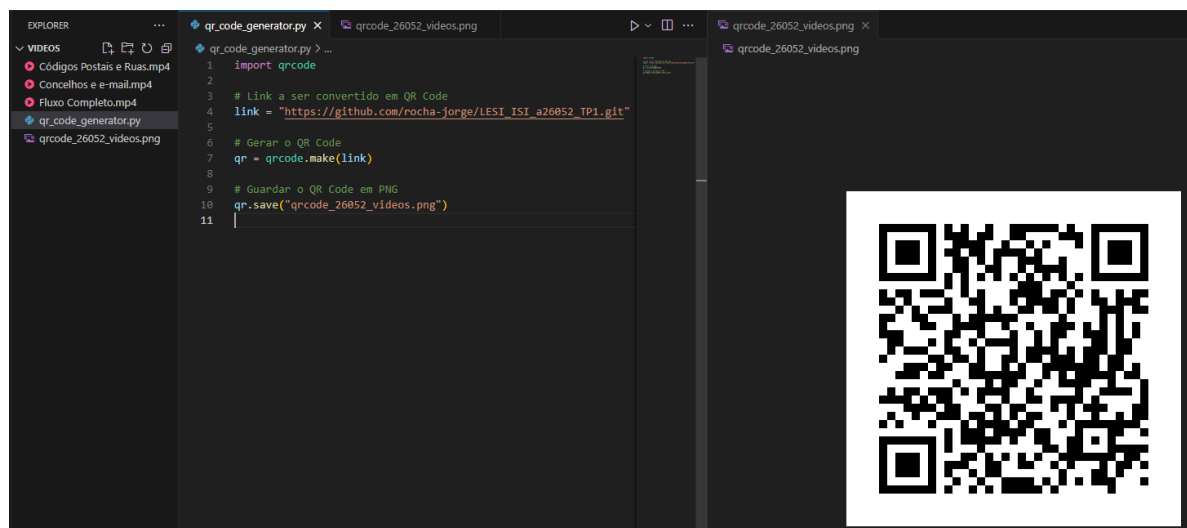
Python package para gerar imagens QR Code a partir de um link

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ajmr1\OneDrive - Instituto Politécnico do Cávado e do Ave\LESI\3_Ano_1_Semestre\ISI\TP1\Videos> pip install qrcode[pil]
Collecting qrcode[pil]
  Downloading qrcode-8.0-py3-none-any.whl.metadata (17 kB)
Collecting colorama (from qrcode[pil])
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting pillow>=9.1.0 (from qrcode[pil])
  Downloading pillow-11.0.0-cp313-cp313-win_amd64.whl.metadata (9.3 kB)
  Downloading pillow-11.0.0-cp313-cp313-win_amd64.whl (2.6 MB)
    2.6/2.6 MB 19.7 MB/s eta 0:00:00
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
  Downloading qrcode-8.0-py3-none-any.whl (45 kB)
Installing collected packages: pillow, colorama, qrcode
Successfully installed colorama-0.4.6 pillow-11.0.0 qrcode-8.0
PS C:\Users\ajmr1\OneDrive - Instituto Politécnico do Cávado e do Ave\LESI\3_Ano_1_Semestre\ISI\TP1\Videos>

```

Código para gerar o QR Code e resultado.



## 11 Conclusão

A realização deste trabalho permitiu entender e experimentar os desafios e potencialidades dos processos de ETL.

São ferramentas incríveis na obtenção, tratamento, circulação, armazenamento, análise e até de visualização de informação.

Uma empresa que dedique os devidos recursos ao estabelecimento de processos de ETL consegue aumentar muito significativamente o seu nível de competitividade. É bastante palpável a da redução de trabalho manual, moroso e repetitivo, sujeito a erro ou a desleixo, e a execução de trabalho que requer grande capacidade de processamento com base em dados por vezes dispersos. A qualidade e atualização dos dados é também facilmente aumentada em várias ordens de grandeza.

Em suma, o presente trabalho permitiu ao autor entender a utilidade e até necessidade, experimentar e estar preparado para encarar um futuro projeto de ETL.

## **12 Trabalhos futuros**

No futuro seria útil implementar processos de ETL que permitissem atualizar, analisar e reportar dados e estatísticas relacionadas com clientes, volume de tráfego, fatores financeiros, entre outros.

Seria também benéfico aumentar o número de controlos de erro e a sua capacidade de informarem os colaboradores da ocorrência e tipo de erros.

Em alguns passos são também aplicados filtros que retiram dados ao longo do processo. Esses dados deveriam ser todos guardados e identificados, e não apenas uma parte deles.

## 13 Referências bibliográficas

<https://www.xweather.com/docs/weather-api>

<https://knime.learnupon.com/enrollments/236128072/details>

<https://hub.knime.com/knime/spaces/Beginners%20Space/~Ln1fgQnWKKeRceeP/>

<https://www.knime.com/knimepress>

<https://www.knime.com/getting-started-guide>

<https://www.youtube.com/user/KNIMETV>

[https://appserver2.ctt.pt/feapl\\_2/app/open/postalCodeSearch/postalCodeSearch.jsp?lang=def](https://appserver2.ctt.pt/feapl_2/app/open/postalCodeSearch/postalCodeSearch.jsp?lang=def)

[https://github.com/centraldedados/codigos\\_postais/blob/master/data/distritos.csv](https://github.com/centraldedados/codigos_postais/blob/master/data/distritos.csv)



## 14 Anexo A – Certificado “Basic Proficiency in KNIME Analytics Platform”

