

Licenciatura em Engenharia de Sistemas Informáticos

ETL de API do IPMA

e comunicação dos dados por e-mail

No âmbito da UC de Integração de Sistemas Informáticos

André Cardoso & Leonel Fernandes
18848 & 18850

Barcelos, 30 de dezembro de 2021

Contents

1	Contextualização	5
2	Introdução	7
3	Extração e Transformação	9
4	Apresentação dos Dados	29
5	Conclusão	37

1 Contextualização

Em contexto da Unidade Curricular de Integração de Sistemas Informáticos é proposto encontrar e desenvolver um problema a resolver através do processo de ETL.

O problema em questão deve incorporar processos como, por exemplo:

- Auditorias:
 - Dados, Processos, Segurança, etc.
- Migração e Reorganização de Dados
- Análise e Processamento de Dados (Datamining, etc.)
- Recomendações e Previsões sobre Estados com Processamento Recorrente a Big Data

Palavras-chave - ETL (Extract Transform Load)

2 Introdução

Este trabalho prático requer que seja encontrado um problema e este mesmo seja resolvido como processo de ETL.

Assim sendo, o tema escolhido foi a organização e análise estatística de dados provenientes da API do IPMA (<https://api.ipma.pt>), que contém dados meteorológicos relativos a cada concelho, assim como as previsões do tempo para cada distrito. No entanto, devido à forma como estão organizados os dados na API, decidimos apenas fazer o processo de ETL para dados do distrito e concelhos de Braga.

Os pontos que queremos abordar neste processo são os seguintes:

- Apresentação dos concelhos com **temperatura máxima** mais elevada nos últimos 10 dias, em cada dia.
- Apresentação dos concelhos com **temperatura mínima** mais baixa nos últimos 10 dias, em cada dia.
- Apresentação dos concelhos com **precipitação máxima** mais elevada nos últimos 10 dias, em cada dia.
- Apresentação das **previsões de tempo** nos próximos 5 dias no concelho de Braga.

Para tratar do processo de ETL foram utilizadas duas ferramentas, **KNIME** e **SPOON**. Estas ferramentas serviram para extração dos dados da API do IPMA, assim como, transformação e agregação dos mesmos para análise estatística.

Para apresentação dos dados foram utilizadas três plataformas diferentes **email**, **Dashboard em C#** e ainda **Discord API**, sendo o último uma plataforma de comunicação muito semelhante à plataforma **Slack**.

O envio de emails deve ter em conta uma lista de consumidores deste serviço que estará armazenada numa base de dados alojada num serviço **Azure**.

Esquema do funcionamento do Processo

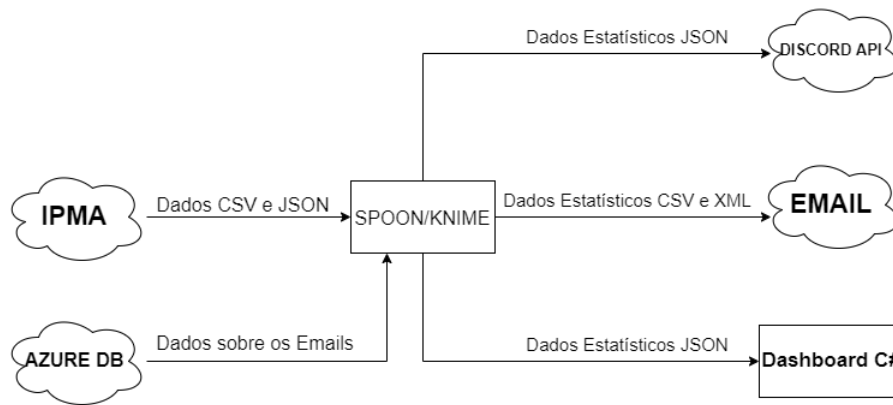


Figure 2.1: Esquema do processo de ETL

Palavras-chave

- **API** (Application Programming Interface)
- **IPMA** (Instituto Português do Mar e da Atmosfera)

3 Extração e Transformação

Como referido no capítulo anterior, para realização da extração e transformação dos dados para análise estatística foram utilizadas as plataformas **KNIME** e **SPOON**. Cada um dos elementos do grupo fez uso de uma das ferramentas para realização do mesmo processo.

Assim sendo, passaremos à explicação das soluções para cada uma das ferramentas utilizadas.

KNIME

A solução desenvolvida em **KNIME** é constituída por 4 passos principais:

- Obtenção dos dados da API.
- Transformação dos dados de forma relevante (dados estatísticos).
- Formatação dos dados em **JSON**, **XML** e **CSV**.
- Envio dos dados para Email.
- Repetição do Processo.

Obtenção dos Dados da API

Na API do IPMA encontram-se disponíveis dados no formato **CSV** relativos a **Temperaturas Máximas**, **Temperaturas Mínimas** e **Dados de Precipitação**. Estes ficheiros encontram-se separados por Concelho o que faz com que seja necessário o acesso ao ficheiro referente a cada concelho individualmente.

Ainda na mesma API estão presentes ficheiros **JSON** referentes à **Previsão do Tempo nos próximos 5 dias**, assim como os ficheiros CSV estes encontram-se separados, neste caso não por concelho mas por Distrito.

Como todos os ficheiros se encontram separados foram criados 4 *workflows*, um para cada tipo de dados necessários, temperaturas máximas e mínimas, precipitação e ainda previsão do tempo.

Para cada um dos concelhos foi necessário utilizar um nodo de leitura de CSV para obtenção dos dados, assim como associada a cada tabela uma coluna extra com o nome de cada local, para ser mais facilmente identificado assim que todos os dados fossem juntos na mesma tabela.

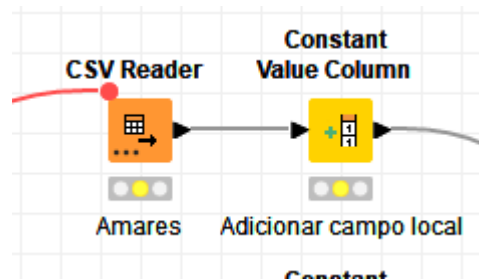


Figure 3.1: Processo de Obtenção de Dados

Como para o Distrito de Braga existem 14 Concelhos, seriam necessários 14 nodos para obter os dados referentes a cada um.

Para cada um dos dados necessários, Temperaturas Máximas e Mínimas e Precipitação a API fornece ficheiro CSV nos seguintes formatos:

Precipitação Total

Row ID	S date	D minimum	D maximum	D range	D mean	D std
Row0	2021-10-01	0	0.002	0.002	0.001	0.001
Row1	2021-10-02	9.606	10.296	0.69	9.858	0.178
Row2	2021-10-03	21.572	25.277	3.706	24.073	0.986
Row3	2021-10-04	2.167	2.795	0.628	2.524	0.175
Row4	2021-10-05	6.566	6.738	0.172	6.686	0.033
Row5	2021-10-06	0.065	0.1	0.034	0.088	0.009
Row6	2021-10-07	0	0	0	0	0
Row7	2021-10-08	0	0	0	0	0
Row8	2021-10-09	0	0.01	0.009	0.003	0.002
Row9	2021-10-10	0	0.007	0.007	0.003	0.003

Figure 3.2: Formato dos dados de Precipitação do IPMA

Temperatura Máxima

Row ID	S date	D minimum	D maximum	D range	D mean	D std
Row0	2021-10-01	25.015	25.121	0.106	25.059	0.021
Row1	2021-10-02	21.057	21.103	0.046	21.09	0.009
Row2	2021-10-03	19.001	19.254	0.253	19.074	0.064
Row3	2021-10-04	17.858	19.993	2.134	19.346	0.586
Row4	2021-10-05	23.901	24.096	0.196	24.01	0.053
Row5	2021-10-06	25.984	26.792	0.807	26.528	0.215
Row6	2021-10-07	27.561	28.687	1.126	28.249	0.295
Row7	2021-10-08	27.037	27.784	0.747	27.454	0.187
Row8	2021-10-09	25.013	25.898	0.885	25.414	0.156
Row9	2021-10-10	25.619	27.594	1.975	27.036	0.544

Figure 3.3: Formato dos dados de Temperatura Máxima do IPMA

Temperatura Mínima

Row ID	S date	D minimum	D maximum	D range	D mean	D std
Row0	2021-10-01	11.749	11.8	0.051	11.786	0.013
Row1	2021-10-02	17.026	17.397	0.37	17.26	0.103
Row2	2021-10-03	9.858	10.1	0.242	10.042	0.06
Row3	2021-10-04	9.436	10.698	1.262	10.303	0.355
Row4	2021-10-05	12.696	12.999	0.303	12.916	0.078
Row5	2021-10-06	10.635	11.412	0.777	10.845	0.185
Row6	2021-10-07	9.732	11.381	1.649	10.485	0.452
Row7	2021-10-08	10.722	12.072	1.35	11.307	0.366
Row8	2021-10-09	11.705	12.396	0.691	11.869	0.175
Row9	2021-10-10	9.429	11.465	2.036	10.284	0.562

Figure 3.4: Formato dos dados de Temperatura Mínima do IPMA

Previsão do Tempo

No entanto, para os dados de previsão do tempo a API fornece dados no formato JSON, dados estes que são convertidos em tabela para serem mais facilmente manuseados, o formato é o seguinte:

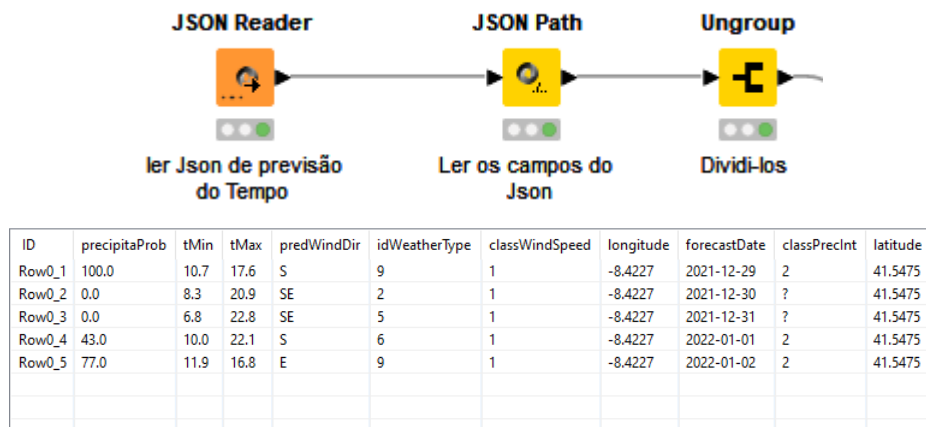


Figure 3.5: Formato dos dados de Previsão do tempo do IPMA

Para cada uma das tabelas acima, à exceção da previsão do tempo, era necessário adicionar um campo que identificasse o local de onde eram provenientes os dados, pois como os ficheiros se encontravam separados esta identificação não existia.

Desta forma, foi adicionada a cada tabela de dados a coluna **"local"** para melhor identificar os dados para análise.

O formato de todas as tabelas seria semelhante a este:

ID	date	minimum	maximum	range	mean	std	local
Row0	2021-10-01	11.7489070892	11.8001489639	0.0512418746948	11.7858477345	0.0133360071035	Amares
Row1	2021-10-02	17.0263061523	17.3965969086	0.370290756226	17.2599304576	0.102654298988	Amares
Row2	2021-10-03	9.85805988312	10.0999250412	0.241865158081	10.0418493601	0.0602232531778	Amares
Row3	2021-10-04	9.43553733826	10.697728157	1.26219081879	10.3026143886	0.355211983857	Amares
Row4	2021-10-05	12.6959505081	12.9993190765	0.30336856842	12.9164590718	0.0777608014015	Amares
Row5	2021-10-06	10.6351737976	11.4120330811	0.776859283447	10.8451588949	0.185109200822	Amares
Row6	2021-10-07	9.7317237854	11.3808088303	1.64908504486	10.4846604901	0.452016138744	Amares
Row7	2021-10-08	10.7219400406	12.071937561	1.34999752045	11.3069415034	0.365640531908	Amares

Figure 3.6: Formato dos dados depois de ser adicionado o campo local

Transformação dos dados de forma relevante

Após a agregação dos dados em tabelas, seria necessário efetuar as devidas transformações de modo a obter os dados estatísticos relevantes. Para cada um dos *workflows*

existem transformações semelhantes de forma a apresentar os dados organizados.

No entanto existem algumas diferenças entre eles, fazendo assim com que seja necessário que sejam apresentados individualmente, principalmente nos dados referentes às previsões do tempo.

Temperatura Máxima/Mínima e Precipitação

O formato genérico para as transformações do *workflow* das **temperaturas e de precipitação** é o seguinte:

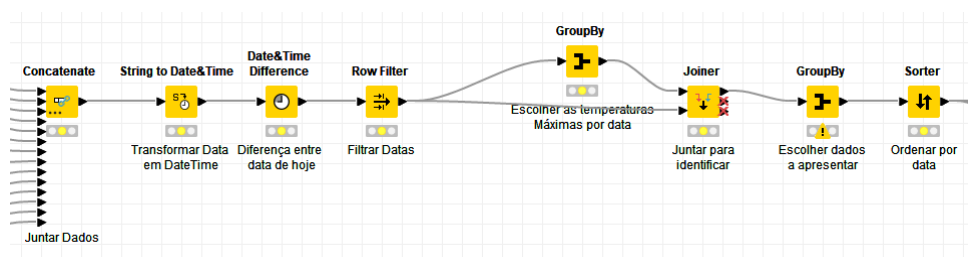


Figure 3.7: Transformações Temperaturas Máxima/Mínima e Precipitação

Após todas as tabelas serem concatenadas no nodo **Concatenate** é calculada a diferença entre a data referente ao dia presente na tabela com o dia atual com o nodo **Date&Time Difference**, este valor é associado a uma coluna que é adicionada à tabela com o nome "diff".

Como apenas queremos apresentar os dados referentes aos **últimos 10 dias** são filtrados os dados cuja diferença seja maior do que 10 através do nodo **Row Filter**

ID	date	minimum	maximum	range	mean	std	local	date(DateTime)	diff
Row76	2021-12-19	1.4174489975	3.33454227448	1.91709327698	2.03551417957	0.51959864128	Amares	2021-12-19	10
Row77	2021-12-20	9.00771713257	9.79744815826	0.789731025696	9.47539760448	0.153704857334	Amares	2021-12-20	9
Row78	2021-12-21	10.0182132721	10.6289577484	0.610744476318	10.2638665305	0.13536036274	Amares	2021-12-21	8
Row79	2021-12-22	11.1616697311	12.0870161057	0.925346374512	11.8627845623	0.243338801267	Amares	2021-12-22	7
Row80	2021-12-23	10.8303127289	12.7049694061	1.87465667725	12.2463475098	0.509198008123	Amares	2021-12-23	6
Row81	2021-12-24	10.2198028564	12.3861923218	2.16638946533	11.6451737086	0.586989697563	Amares	2021-12-24	5
Row82	2021-12-25	9.88567066193	10.9021692276	1.01649856567	10.6462831262	0.282942699307	Amares	2021-12-25	4
Row83	2021-12-26	10.6403446198	11.8000659943	1.15972137451	11.5018741584	0.322551771394	Amares	2021-12-26	3

Figure 3.8: Tabela com a coluna "diff" associada

Depois de filtrados os dados são organizados pelo valor que nós queremos obter por cada data através do nodo **Group By**.

Este nodo organiza os dados por data, e faz a sua agregação mediante um valor, no caso da figura acima agrega os dados pela maior temperatura encontrada com uma data associada

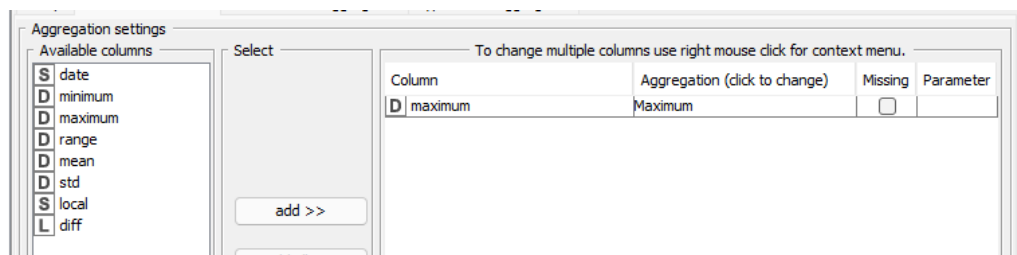


Figure 3.9: Agregação valor máximo de temperatura

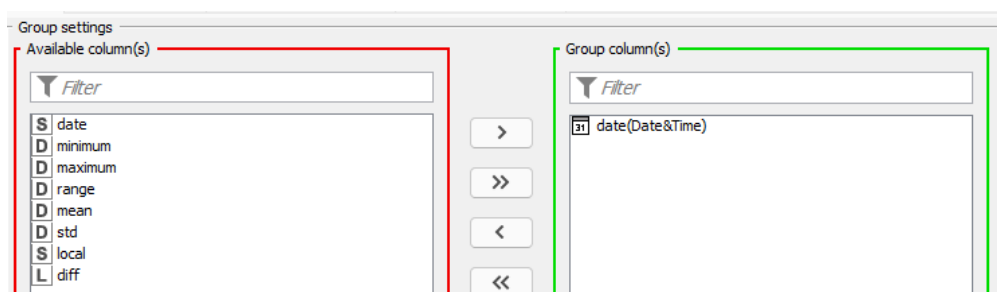


Figure 3.10: Organização por Data

Depois de organizados, os dados são ordenados por data através do nodo **Sorter**

Este processo é repetido nos 3 processos (Temperatura Máxima/Mínima e Precipitação), no entanto, no processo de temperatura mínima, os dados ao invés de organizados pelo maior valor, são organizados pelo menor.

Previsão do Tempo

Para o processo de transformação de dados de previsão do tempo, o *workflow* ocorre de forma diferente.

O formato dos dados de previsão do tempo inclui diversos **ID** para dados dos quais não temos acesso direto, para termos acesso a tais dados temos de associar os **ID** a tabelas presentes na API com esse propósito.

O tipo de dados dos quais não temos acesso são: **Classe de Precipitação**, **Classe da velocidade do Vento** e ainda o **Tipo de Clima**.

A tabela sem qualquer tipo de transformações encontra-se da seguinte forma:

ID	precipitaProb	tMin	tMax	predWindDir	idWeatherType	classWindSpeed	longitude	forecastDate	classPrecInt	latitude
Row0_1	100.0	10.7	17.6	S	9	1	-8.4227	2021-12-29	2	41.5475
Row0_2	0.0	8.3	20.9	SE	2	1	-8.4227	2021-12-30	?	41.5475
Row0_3	0.0	6.8	22.8	SE	5	1	-8.4227	2021-12-31	?	41.5475
Row0_4	43.0	10.0	22.1	S	6	1	-8.4227	2022-01-01	2	41.5475
Row0_5	77.0	11.9	16.8	E	9	1	-8.4227	2022-01-02	2	41.5475

Figure 3.11: Tabela de Previsão do Tempo sem transformações

As tabelas às quais temos de associar a tabela inicial, com dados referentes ao tipo de clima, precipitação e velocidade do vento são as seguintes:

ID	descClassWindSpeedDailyEN	descClassWindSpeedDailyPT	classWindSpeed
Row0_1	--	---	-99
Row0_2	Weak	Fraco	1
Row0_3	Moderate	Moderado	2
Row0_4	Strong	Forte	3
Row0_5	Very strong	Muito forte	4

Figure 3.12: Tabelas de descrição da velocidade do Vento

ID	descClassPrecIntEn	descClassPrecIntPT	classPrecInt
Row0_1	--	---	-99
Row0_2	No precipitation	Sem precipitação	0
Row0_3	Weak	Fraco	1
Row0_4	Moderate	Moderado	2
Row0_5	Strong	Forte	3

Figure 3.13: Tabela de descrição de Precipitação

ID	descIdWeatherTypeEN	descIdWeatherTypePT	idWeatherType
Row0_1	--	---	-99
Row0_2	No information	Sem informação	0
Row0_3	Clear sky	Céu limpo	1
Row0_4	Partly cloudy	Céu pouco nublado	2
Row0_5	Sunny intervals	Céu parcialmente nublado	3
Row0_6	Cloudy	Céu muito nublado ou encoberto	4
Row0_7	Cloudy (High cloud)	Céu nublado por nuvens altas	5
Row0_8	Showers	Aguaceiros	6
Row0_9	Light showers	Aguaceiros fracos	7

Figure 3.14: Tabela de descrição do Clima

Para cada uma destas tabelas está associado um nodo **Joiner** para juntar as duas tabelas através do **ID** correspondente, assim como um **Column Filter** para remover as colunas não desejadas.

O *workflow* procede da seguinte forma:

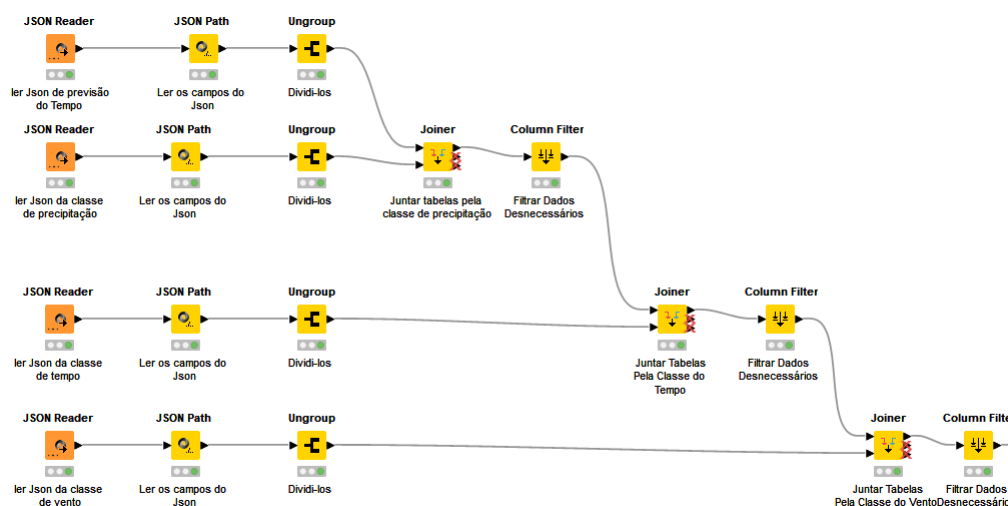


Figure 3.15: Transformações no workflow de previsão de Tempo

Os dados depois de todas as transformações geram uma tabela com o seguinte formato:

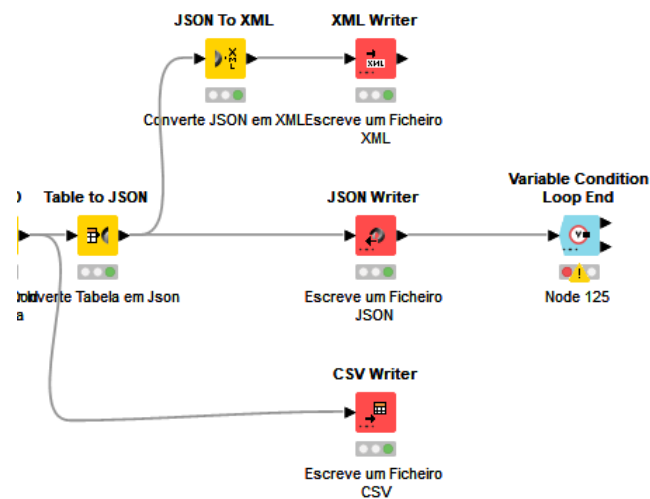
ID	precipitaProb	tMin	tMax	predWindDir	longitude	forecastDate	latitude	descClassPrecIntPT	descIdWeatherTypePT	descClassWindSpeedDailyPT
Row0	0.0	8.3	20.9	SE	-8.4227	2021-12-30	41.5475	?	Céu pouco nublado	Fraco
Row1	0.0	6.8	22.8	SE	-8.4227	2021-12-31	41.5475	?	Céu nublado por nuvens altas	Fraco
Row2	43.0	10.0	22.1	S	-8.4227	2022-01-01	41.5475	Moderado	Aguaceiros	Fraco
Row3	100.0	10.7	17.6	S	-8.4227	2021-12-29	41.5475	Moderado	Chuva	Fraco
Row4	77.0	11.9	16.8	E	-8.4227	2022-01-02	41.5475	Moderado	Chuva	Fraco

Figure 3.16: Tabela final de previsão do tempo

Formatação dos dados em JSON, XML e CSV

Depois do processo anterior é necessário, para cada um dos *workflows*, guardar os dados em diversos formatos de ficheiro, neste caso **JSON**, **XML** e **CSV**.

Para esse efeito foram utilizados os mesmos nodos para transformação nos 3 formatos:



Estes nodos utilizam os dados provenientes das tabelas e guardam ficheiros nas pastas XML, CSV e JSON.

Estes ficheiros podem ser acedidos posteriormente para os processos de apresentação de dados (Email, Dashboard, Discord API).

Envio de Dados para Email

Uma das formas de apresentação dos dados é através do envio diário de um email com as tabelas referentes aos dados relevantes.

Para o envio do email é necessário que o **KNIME** faça o processamento das tabelas de forma a serem legíveis por parte do recipiente.

Formatação das Tabelas

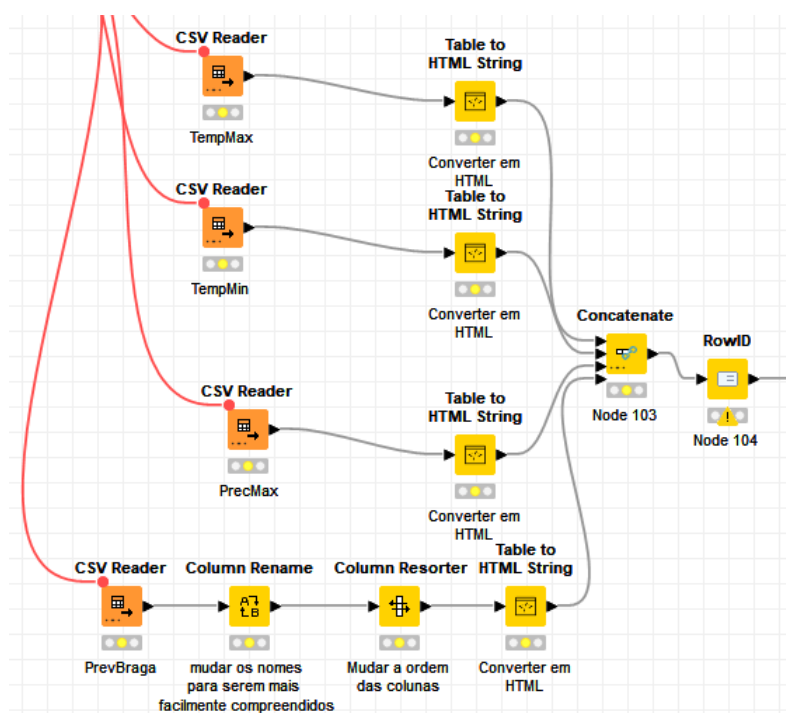


Figure 3.17: Formatação de dados para email

Para as tabelas serem legíveis por parte do recipiente é necessário fazer a formatação para **HTML**, para isso é utilizado o nó **Table to HTML string** que trata de tal formatação. A tabela de previsões no entanto tem dois nós extra que servem para renomear as colunas para serem mais facilmente compreendidas e ainda a sua reordenação para o mesmo efeito.

Depois de convertidas para **HTML** as tabelas são juntas para poderem ser enviadas como variáveis no email.

Envio de email

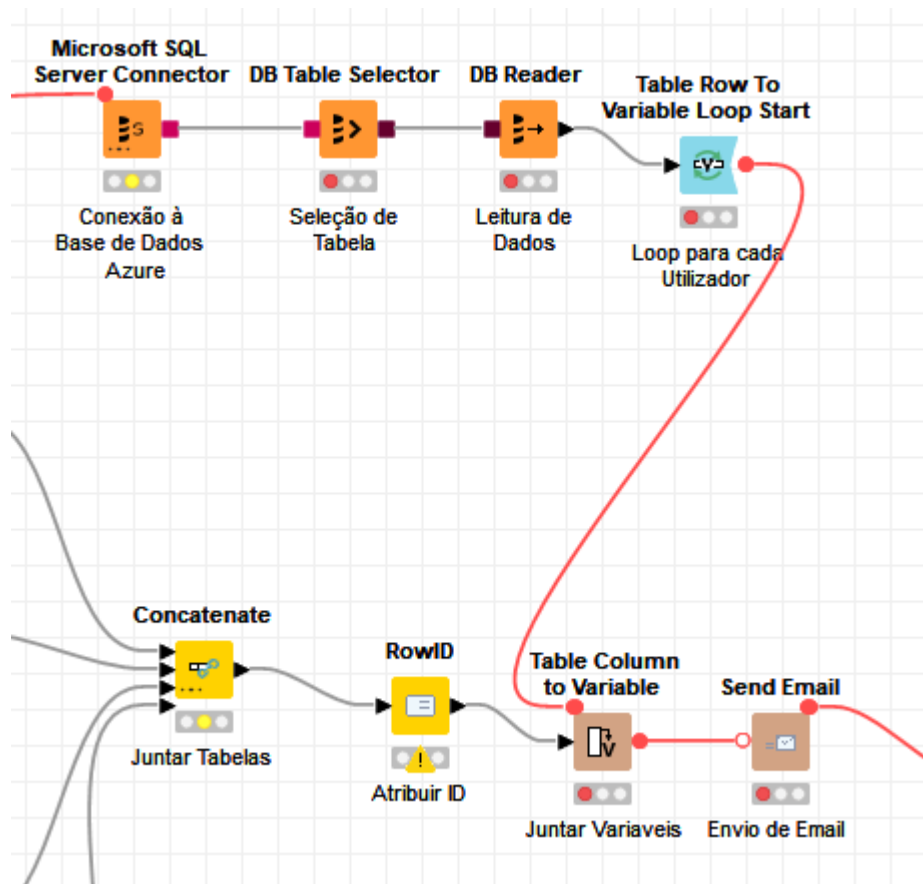


Figure 3.18: Envio de email

Para proceder ao envio dos emails para os utilizadores, o **KNIME** tem de aceder primeiro à base de dados **Azure** que definimos, esta base de dados encontra-se no servidor "isitp2tempo.database.windows.net". Nesta base de dados temos os utilizadores para os quais enviaremos emails.

Os dados presentes na base de dados são os seguintes:

nome	zona	email
Leonel Fernandes	Braga	Leonelsfernandes@outlook.pt
André Cardoso	Aveiro	andrelope.cardoso@gmail.com
Miguel Silva	Sagres	miguel.silva.5901.steam@gmail.com
Leonel Fernandes Aluno	Faro	a18850@alunos.ipca.pt
André Cardoso Aluno	Loulé	a18848@alunos.ipca.pt

Figure 3.19: Lista de utilizadores para envio de email

Após obter os utilizadores o nodo **Table Row To Variable Loop Start** vai repetir o processo de envio para cada um dos presentes na tabela.

A escolha de destinatário também é definida através da variável **email** presente na tabela.

Quando o processo estiver concluído todos os utilizadores receberão um email com as tabelas dos dados meteorológicos proveniente do email "weather.isitp2@outlook.pt".

Repetição do Processo

Todos os processos referidos até agora necessitam ser automatizados para terem algum tipo de utilidade, como tal, foram utilizados nodos disponibilizados na plataforma **KNIME** para criar repetição de todos os processos definindo um intervalo de tempo.

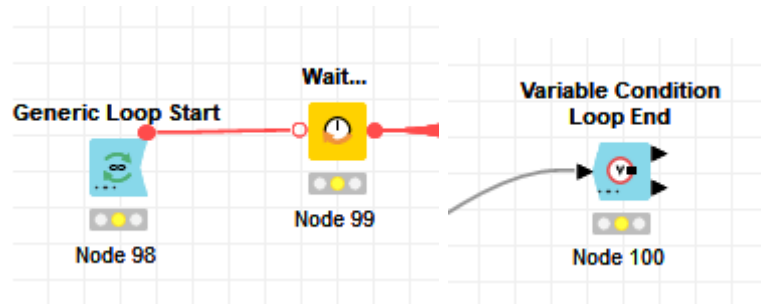


Figure 3.20: Processo de Loop

Para a **obtenção dos dados** foi definido o intervalo de tempo de 1 hora entre cada request dos ficheiros CSV para a API do IPMA, isto é feito para manter os dados atualizados.

Já para o processo de **envio de email** foi definido o intervalo de tempo de 24 horas entre cada iteração do programa.

SPOON

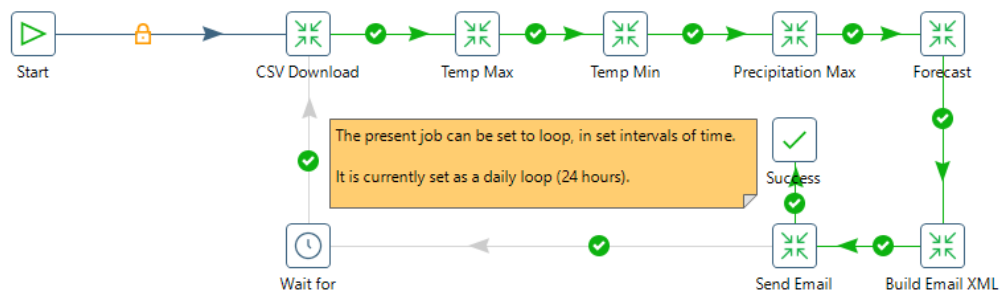


Figure 3.21: Job Desenvolvido

Extração de Informação

Por motivos de performance, o módulo *CSV Input* não permite o carregamento de um ficheiro CSV utilizando um URL como fonte. Assim, de modo a termos acesso à informação necessária, é necessário o download de todos os ficheiros de modo persistente - um total de 14 ficheiros, um por cada concelho do distrito de Braga, disponível na API.

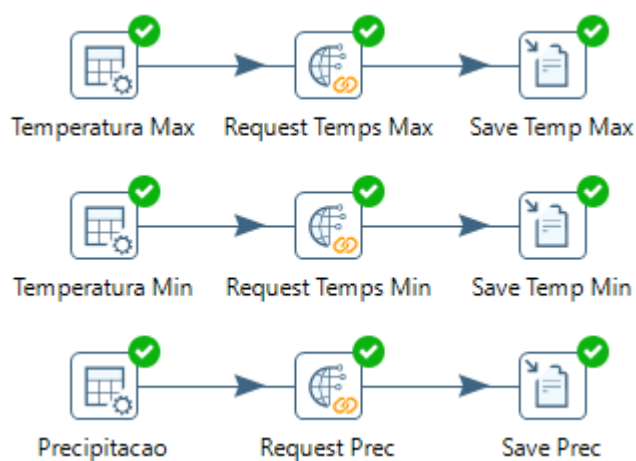


Figure 3.22: Transformação *CSV Download*

Formato dos Ficheiros CSV

Tendo em conta a estrutura idêntica dos ficheiros de **Temperatura Máxima**, **Mínima** e **Precipitação (Máxima)**, o formato para extração dos dados é igualmente válido

para todos.

#	Name	Type	Format
1	date	Date	yyyy-MM-dd
2	minimum	BigNumber	#####.###
3	maximum	BigNumber	#####.###
4	range	BigNumber	#####.###
5	mean	BigNumber	#####.###
6	std	BigNumber	#####.###

Figure 3.23: Formato de Leitura dos Ficheiros CSV descarregados.

#	date	minimum	maximum	range	mean	std	filename
1	2021-10-01	25.015	25.121	0.106	25.059	0.021	amares.csv
2	2021-10-02	21.057	21.103	0.046	21.09	0.009	amares.csv
3	2021-10-03	19.001	19.254	0.253	19.074	0.064	amares.csv
4	2021-10-04	17.858	19.993	2.134	19.346	0.586	amares.csv
5	2021-10-05	23.901	24.096	0.196	24.01	0.053	amares.csv

Figure 3.24: Pré-visualização dos dados dos ficheiros.
Ex.: Temperatura Máxima

A **Previsão** do tempo no entanto, é obtida através de formatação JSON, e pode ser lida através do módulo [JSON Input] utilizando o url correspondente, não sendo necessário o download persistente.

#	Name	Path	Type	Format
1	precipitaProb	\$.[*].precipitaProb	Number	#,#
2	tMin	\$.[*].tMin	Number	#,#
3	tMax	\$.[*].tMax	Number	#,#
4	predWindDir	\$.[*].predWindDir	String	
5	idWeatherType	\$.[*].idWeatherType	Integer	
6	classWindSpeed	\$.[*].classWindSpeed	Integer	
7	longitude	\$.[*].longitude	String	
8	forecastDate	\$.[*].forecastDate	Date	yyyy-MM-dd
9	classPrecInt	\$.[*].classPrecInt	Integer	
10	latitude	\$.[*].latitude	String	

Figure 3.25: Formato de Leitura dos dados JSON. JSON-Path

#	TYPE	precipi...	tMin	tMax	predWi...	idWea...	class...	longitude	forecastDate	classP...	latitude
1	info	100	10.7	17.6	S	9	1	-8.4227	2021-12-29	2	41.5475
2	info	0	8.3	21.1	SE	2	1	-8.4227	2021-12-30	<null>	41.5475
3	info	0	6.4	22.8	SE	5	1	-8.4227	2021-12-31	<null>	41.5475
4	info	46	8.8	22.1	S	6	1	-8.4227	2022-01-01	2	41.5475
5	info	88	10.8	18.6	S	9	1	-8.4227	2022-01-02	2	41.5475

Figure 3.26: Pré-visualização dos dados extraídos.

Transformação dos Dados Extraídos

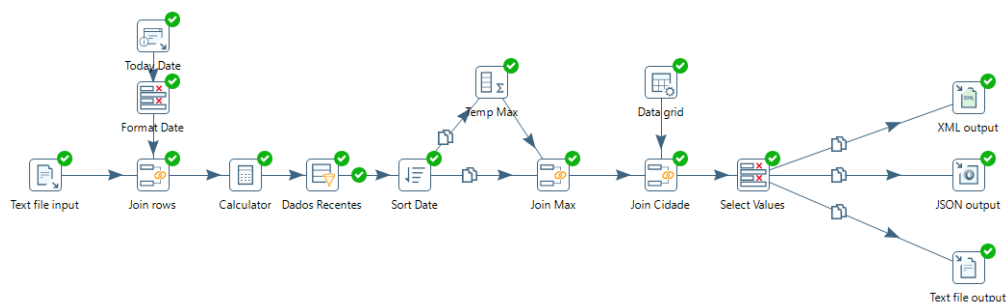


Figure 3.27: Formato Geral das transformações de um CSV.
Válido para Temp. Máx. e Mín., e Prec. Máx.

É utilizado o módulo **[Text file input]** para que se possa carregar facilmente todos os ficheiros CSV. Infelizmente este método utiliza um caminho completo para o diretório, pelo que se revela uma má solução.

É calculada a diferença entre os dias das informações e o dia atual, recorrendo a informações do sistema, e filtrados aqueles que se encontram no intervalo entre 0 e 10 dias de diferença. São de seguida ordenados por Data e, utilizando o módulo **[Group By]**, é possível encontrar as temperaturas Máximas de cada dia.

A nova tabela gerada é cruzada com a tabela anterior de modo a reaver os dados perdidos pela operação de Group By.

Por fim, os nomes dos ficheiros são utilizados para atribuir o nome da Cidade correspondente, selecionam-se os dados com informação relevante direcionando-os para os seus outputs: XML, JSON e CSV.

#	Fieldname	Element name	Content type	Type	Format
1	Data	data	Attribute	Date	yyyy-MM-dd
2	Cidade		Element	String	
3	Maximo		Element	BigNumber	

Figure 3.28: Formato XML

Os outputs JSON e CSV são bastante simples na sua essência, no entanto o output XML permite algumas alterações como a definição do campo **'data'** como um atributo do Elemento **'Day'** do Elemento Raíz **'Temp-Max/Temp-Min/Prec-Max'**.

Para a informação da **Previsão** do tempo, são necessárias 3 rotas com informação das classes de:

- Precipitation - Precipitação
- Wind - Vento
- Weather - Clima

Estas rotas retornam informação em formato JSON relativamente a:

- Quantidade de Precipitação

descClassPrecIntEn	descClassPrecIntPT	classPrecInt
--	---	-99
No precipitation	Sem precipitação	0
Weak	Fraco	1
Moderate	Moderado	2
Strong	Forte	3

- Força de Vento

descClassWindSpeedDaily...	descClassWindSpeedDail...	ClassWindSpeed
--	---	-99
Weak	Fraco	1
Moderate	Moderado	2
Strong	Forte	3
Very strong	Muito forte	4

- Estado Meteorológico

descIdWeatherTypeEN	descIdWeatherTypePT	idWeatherType
--	---	-99
No information	Sem informação	0
Clear sky	Céu limpo	1
Partly cloudy	Céu pouco nublado	2
Sunny intervals	Céu parcialmente nublado	3
Cloudy	Céu muito nublado ou encoberto	4

Esta informação é cruzada com os id's utilizados nas previsões fornecidas, e posteriormente substituídas por completo por descrições do estado. Este processo é demarcado pelos módulos do tipo **[Merge Join]**.

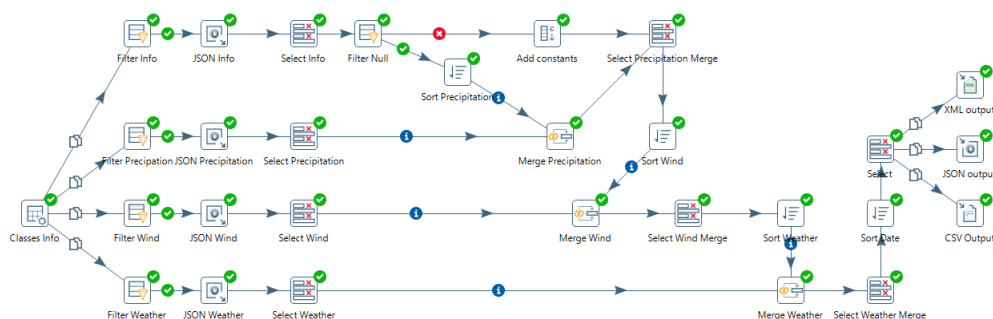


Figure 3.29: Formato XML

Como alguns campos da coluna ‘**classPrecInt**’ podem ser *null*, nesses casos, são adicionadas as colunas da tabela de informações da classe de Precipitação com valores ‘**N/A**’.

Envio de Dados para Email

Utilizando uma base de dados, e uma página HTML construída previamente com as informações a fornecer, são definidos os campos necessários para o envio do email (*sender* email e *password*, *server* e *server port* e ainda o assunto, *subject*, do email).

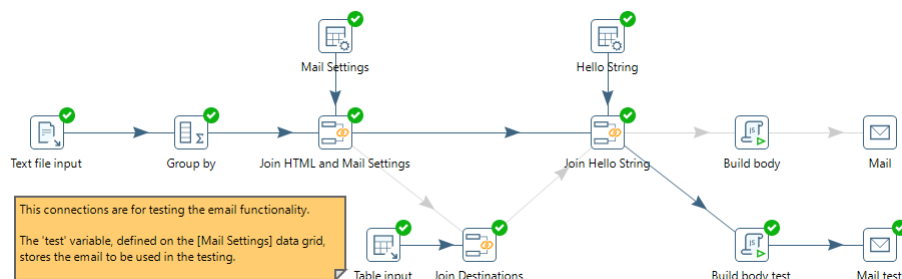


Figure 3.30: Carregamento de página HTML e envio por email.

Construção de página HTML

Criando uma *template* XSL, é possível utilizar um ficheiro XML com a informação desejada para criar uma página HTML com tabelas visuais dos Dados.

Para tal é necessário agregar os dados previamente gerados num único XML.

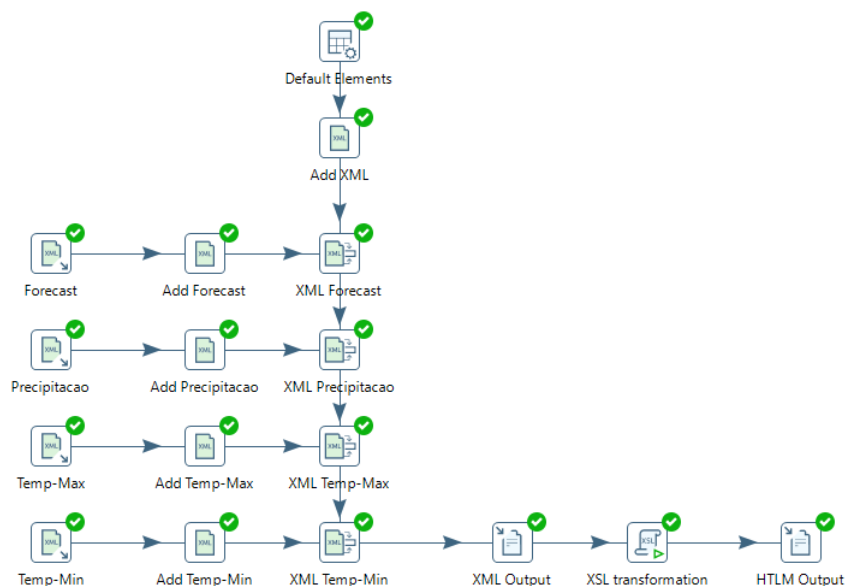


Figure 3.31: Construção de página HTML utilizando XSL.

4 Apresentação dos Dados

Para proceder à apresentação dos dados transformados utilizamos 3 plataformas, **Dashboard em C#**, **Email** e ainda **Discord API**, neste capítulo entraremos em detalhe de como estes processos estão definidos, assim como o seu funcionamento e apresentação.

Email

O processo de apresentação de dados por email é feito maioritariamente através do **SPOON/KNIME**, estes transformam as tabelas de dados em texto **HTML** para poder ser apresentado da seguinte forma:

Previsão dos próximos 5 dias

Data	Temperatura		longitude	latitude	Precipitação	Probabilidade	Vento	Direção	Clima
	Min	Max							
2021-12-29	10.7	17.6	-8.423	41.547	Moderado	100	Fraco	S	Chuva
2021-12-30	8.3	21.1	-8.423	41.547	N/A	0	Fraco	SE	Céu pouco nublado
2021-12-31	6.4	22.8	-8.423	41.547	N/A	0	Fraco	SE	Céu nublado por nuvens altas
2022-01-01	8.8	22.1	-8.423	41.547	Moderado	46	Fraco	S	Aguaceiros
2022-01-02	10.8	18.6	-8.423	41.547	Moderado	88	Fraco	S	Chuva

Precipitação Máxima

Data	Cidade	Máximo
2021-12-28	Braga	5.5
2021-12-27	Cabeceiras de Basto	11.6
2021-12-26	Vizela	37.063
2021-12-25	Vila Nova de Famalicão	6.648
2021-12-24	Terras de Bouro	26.753
2021-12-23	Terras de Bouro	18.063
2021-12-22	Terras de Bouro	3.353
2021-12-21	Braga	10.4
2021-12-20	Braga	28.2

Temperatura Mínima

Data	Cidade	Mínimo
2021-12-24	Cabeceiras de Basto	6.8
2021-12-23	Cabeceiras de Basto	7.7
2021-12-21	Cabeceiras de Basto	9.02
2021-12-28	Celorico de Basto	5.124
2021-12-20	Celorico de Basto	7.701
2021-12-27	Terras de Bouro	10.349
2021-12-26	Terras de Bouro	9.162
2021-12-25	Terras de Bouro	8.602
2021-12-22	Terras de Bouro	9.953

Temperatura Máxima

Data	Cidade	Máximo
2021-12-24	Braga	15.1
2021-12-25	Vila Nova de Famalicão	16.696
2021-12-28	Vila Verde	18.889
2021-12-27	Vila Verde	17.287
2021-12-26	Vila Verde	17.098
2021-12-23	Vila Verde	17.314
2021-12-22	Vila Verde	18.304
2021-12-21	Vila Verde	20.897
2021-12-20	Vila Verde	16.193

Figure 4.1: Email recebido

Dashboard em C#

Para apresentação dos dados foi ainda utilizado um simples Dashboard feito em **Windows Forms** com 4 tabelas, uma para cada tipo de dados transformados.

O **Dashboard** tem o seguinte formato:

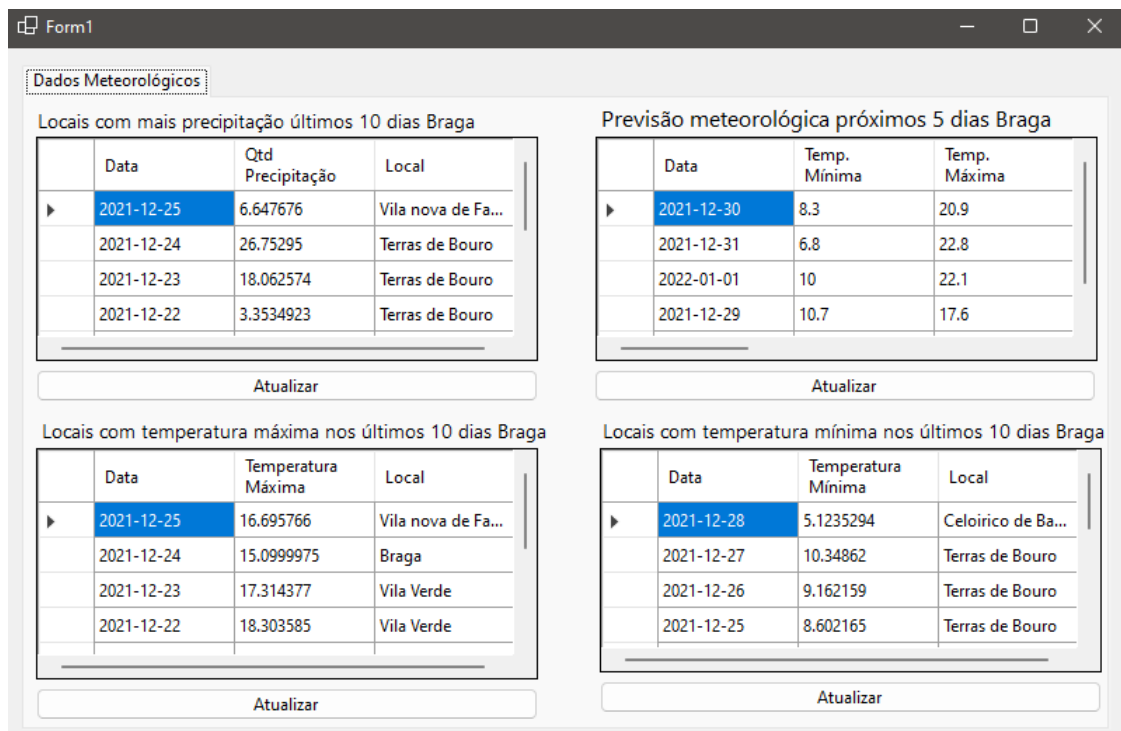


Figure 4.2: Dashboard

O programa utiliza os ficheiros **JSON** guardados depois do processo de transformação de dados e converte os mesmo para as devidas *Class* no *C#*.

As classes utilizadas são as seguintes:

```

2 references
class TempMax
{
    [System.ComponentModel.DisplayName("Data")]
    0 references
    public string date { get; set; }
    [System.ComponentModel.DisplayName("Temperatura Máxima")]
    0 references
    public float maximum { get; set; }
    [System.ComponentModel.DisplayName("Local")]
    0 references
    public string local { get; set; }
}

```

Figure 4.3: *Class* de Temperatura Máxima

```

2 references
class TempMin
{
    [System.ComponentModel.DisplayName("Data")]
    0 references
    public string date { get; set; }
    [System.ComponentModel.DisplayName("Temperatura Mínima")]
    0 references
    public float minimum { get; set; }
    [System.ComponentModel.DisplayName("Local")]
    0 references
    public string local { get; set; }
}

```

Figure 4.4: *Class* de Temperatura Mínima

```

2 references
public class itemPrec
{
    [System.ComponentModel.DisplayName("Data")]
    0 references
    public string date { get; set; }
    [System.ComponentModel.DisplayName("Qtd Precipitação")]
    0 references
    public float maximum { get; set; }
    [System.ComponentModel.DisplayName("Local")]
    0 references
    public string local { get; set; }
}

```

Figure 4.5: *Class* de Precipitação Máxima


```

2 references
class PrevBraga
{
    [System.ComponentModel.DisplayName("Data")]
    0 references
    public string forecastDate { get; set; }
    [System.ComponentModel.DisplayName("Temp. Mínima")]
    0 references
    public float tMin { get; set; }
    [System.ComponentModel.DisplayName("Temp. Máxima")]
    0 references
    public float tMax { get; set; }
    [System.ComponentModel.DisplayName("Descrição")]
    0 references
    public string descIdWeatherTypePT { get; set; }
    [System.ComponentModel.DisplayName("Prob. Precipitação")]
    0 references
    public float precipitaProb { get; set; }
    [System.ComponentModel.DisplayName("Tipo de Precipitação")]
    0 references
    public string descClassPrecIntPT { get; set; }
    [System.ComponentModel.DisplayName("Velocidade do Vento")]
    0 references
    public string descClassWindSpeedDailyPT { get; set; }
    [System.ComponentModel.DisplayName("Direção do Vento")]
    0 references
    public string predWindDir { get; set; }
    0 references
    public float longitude { get; set; }
    0 references
    public float latitude { get; set; }
}

```

Figure 4.6: *Class* de Previsão de Tempo

Depois de guardados os dados como cada uma das classes, estes são apresentados no devido **DataGridView** do **Dashboard**.

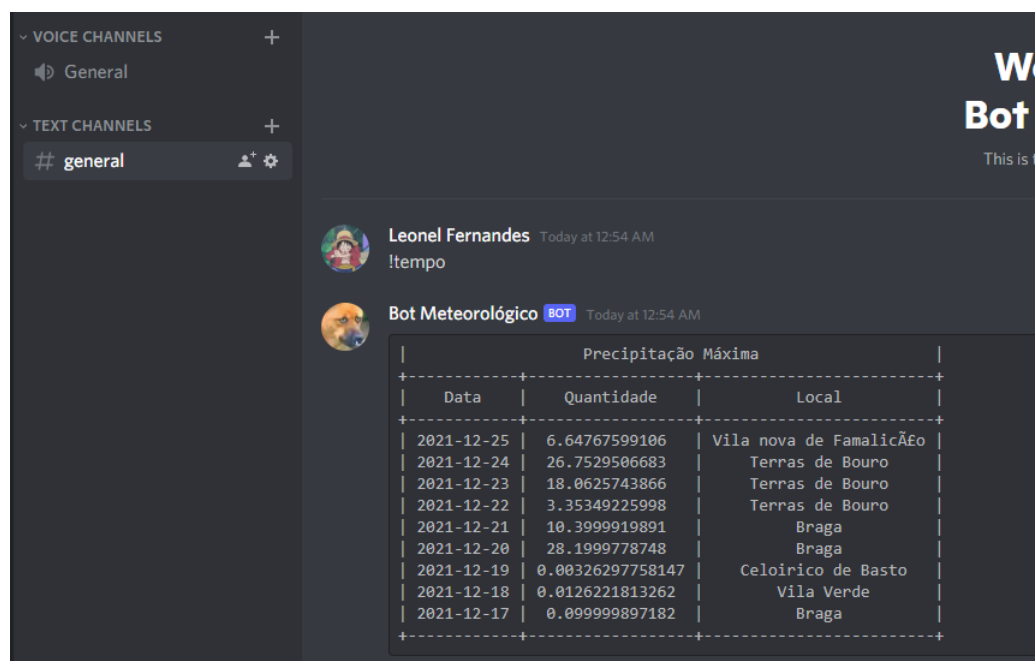
Discord API

Discord é uma plataforma de comunicação onde é possível enviar mensagens para canais de texto, assim como falar em canais de voz, esta plataforma em muito se assemelha à plataforma **Slack**.

Assim como na plataforma Slack, o Discord disponibiliza uma API onde é possível efetuar diversas ações nos canais de comunicação. Como por exemplo a solução criada na execução deste trabalho prático, um **BOT** de envio de mensagens sobre dados meteorológicos. Para utilização da API é-nos disponibilizado um **token** de acesso com o qual fazemos a comunicação.

O **BOT** foi desenvolvido em **python** pois é disponibilizada uma biblioteca que em muito facilita a comunicação com a **Discord API**.

Assim que temos o processo do BOT a correr ele espera por um comando específico para ir buscar os dados **JSON** aos ficheiros e enviar para o utilizador, neste caso o comando é **"!tempo"**. Após receber o comando o BOT envia tabelas formatadas em ASCII referentes aos dados para o canal de texto.



The screenshot shows a Discord chat window with a sidebar on the left containing 'VOICE CHANNELS' and 'TEXT CHANNELS'. The 'general' text channel is selected. In the chat area, a user named 'Leonel Fernandes' has sent the command '!tempo'. The bot 'Bot Meteorológico' has responded with a table of precipitation data.

Precipitação Máxima		
Data	Quantidade	Local
2021-12-25	6.64767599106	Vila nova de Famalicão
2021-12-24	26.7529506683	Terras de Bouro
2021-12-23	18.0625743866	Terras de Bouro
2021-12-22	3.35349225998	Terras de Bouro
2021-12-21	10.3999919891	Braga
2021-12-20	28.1999778748	Braga
2021-12-19	0.00326297758147	Celoirico de Basto
2021-12-18	0.0126221813262	Vila Verde
2021-12-17	0.099999897182	Braga

Figure 4.7: Resposta do BOT do Discord

O BOT apresenta todas as 4 tabelas referentes aos dados. Estas tabelas são formatadas utilizando uma biblioteca do python chamada "**prettytable**".

```
#Obter dados de locais de precipitação Máxima e formatar em tabela
def getPrecMaxTable():

    path = os.getcwd()
    path = path.replace("DiscordBot", "")
    path = path + "KNIME\\Json"

    precMaxFile = open(str(path) + "\\File_0PrecMax.json")
    precMaxData = json.load(precMaxFile)
    precMaxTable = PrettyTable(["Data", "Quantidade", "Local"])
    precMaxTable.title = 'Precipitação Máxima'

    for element in precMaxData:
        precMaxTable.add_row([element['date'], element['maximum'], element['local']])

    return precMaxTable
```

Figure 4.8: Código de formatação de Tabela em Python

Assim como referido anteriormente, o programa tem como fonte de dados os ficheiros **JSON** para cada tabela.

Depois de formatado apenas envia a tabela como mensagem pelo **Discord**.

```
@client.event
async def on_message(message):
    if message.author == client.user:
        return

    msg = message.content

    if msg.startswith("!tempo"):
        precMaxTable = dados.getPrecMaxTable()
        tempMaxTable = dados.getTempMaxTable()
        tempMinTable = dados.getTempMinTable()
        prevTable = dados.getPrevTable()

        await message.channel.send("````" + str(precMaxTable) + "````")
        await message.channel.send("````" + str(tempMaxTable) + "````")
        await message.channel.send("````" + str(tempMinTable) + "````")
        await message.channel.send("````" + str(prevTable) + "````")
```

Figure 4.9: Envio de Mensagem para o Discord

5 Conclusão

O trabalho prático tinha como objetivo a definição e resolução de um problema de ETL através de ferramentas com esse mesmo propósito.

Foi definido como problema a análise estatística de dados provenientes da API do IPMA e a sua consequente apresentação. As estatísticas apresentadas são:

- Concelhos com temperatura mais elevada organizados por data por data nos últimos 10 dias.
- Concelhos com temperatura mais baixa organizados por data por data nos últimos 10 dias.
- Concelhos com mais precipitação organizados por data nos últimos 10 dias.
- Previsão do tempo para os próximos 5 dias.

Como tal escolhemos 2 plataformas para desenvolvimento do processo de análise, o KNIME e o SPOON. Os dados depois de analisados são guardados em 3 formatos, XML, CSV e JSON, para serem posteriormente utilizados nas plataformas escolhidas para apresentação.

As plataformas que escolhemos para apresentação dos dados foram, Email, Dashboard em C# e ainda Discord, todas estas plataformas fazem uso dos ficheiros gerados em KNIME/SPOON. Para o envio de Email decidimos utilizar a plataforma Azure SQL para guardar os dados dos utilizadores.

Escolhemos apenas apresentar dados referentes ao distrito de Braga e aos seu concelhos devido à forma como estes estão organizados na API, a apresentação dos distritos restantes seria um processo demorado. Seria uma mais valia o desenvolvimento desta solução contendo dados para todos os distritos, apesar do esforço extra requerido.

No envio de email também seria uma mais valia ser desenvolvido o processo de envio de email para os utilizadores de cada região com os dados da mesma. Assim como foi referido anteriormente, isto não foi desenvolvido pela forma como estes dados estão organizados na API.

Como nota final achamos que tudo o que foi pedido para este trabalho prático foi desenvolvido, desde o processo de ETL até apresentação dos dados. Penso que existem algumas melhorias a serem postas em prática, as mesmas foram referidas nos dois parágrafos acima. De futuro seria uma mais valia a resolução dos problemas referidos.