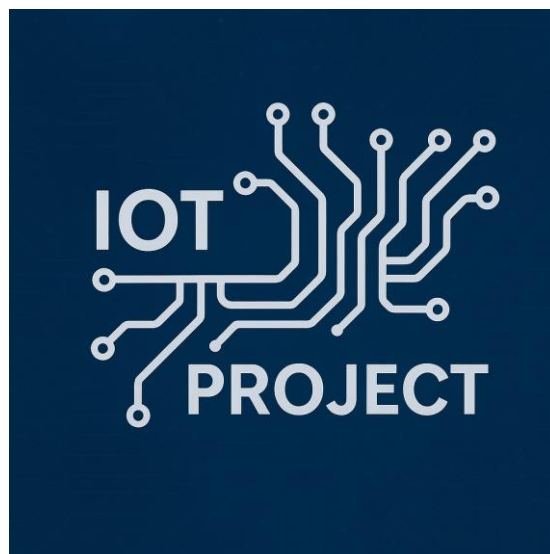




Instituto Politécnico do Cávado e do Ave
Escola Superior de Tecnologia
Engenharia de Sistemas Informáticos



Integração de Sistemas Informação
Aplicação do Processo ETL:
Ánalse de dados Meteorológicos
Trabalho Prático

Docente: Óscar Ribeiro

Alunos: Maria Eduarda Checan

Data: 18/10/2025

Índice

I.	Introdução	4
II.	Problema.....	5
2.1	Estrutura da solução.....	5
2.2	Visão Geral do Funcionamento	7
III.	Estratégia Utilizada	8
3.1	Extração	8
3.2	Transformação	12
3.3	Carregamento	14
IV.	Jobs.....	18
4.1	Job 1 — ETL_IoT_API.....	18
4.2	Job 2 — LOG Manager.....	18
4.3	Job 3 — Success Flow and Email Notification.....	19
V.	Vídeo com demonstração (QR Code)	20
VI.	Conclusão e Trabalhos Futuros	21
VII.	Referências Bibliográficas.....	22

Índice Imagens

Figura 1 – OpenWeather API.....	9
Figura 2 - Fluxo Node-RED.....	9
Figura 3 - Lógica Node-RED	10
Figura 4 - Dados Simulados.....	10
Figura 5 - Leitura Dados Simulados	11
Figura 6 - Leitura Dados API	11
Figura 7 - Joiner	12
Figura 8 - Dados API Transformados.....	13
Figura 9 - Dados IoT Transformados	13
Figura 10 - Dados MySQL.....	14
Figura 11 - Dados CSV Joiner	15
Figura 12 - Dados CSV das Estatísticas geradas em C#.....	15
Figura 13 - Dados CSV API	16
Figura 14 - PowerBI.....	16
Figura 15 - Dados XML gerado em C#.....	17
Figura 16 - Ficheiro CSV dos Logs.....	19
Figura 17 - Notificação de Email após conclusão do fluxo ETL.....	20
Figura 18 - QR Code.....	20

I. Introdução

No âmbito da Unidade Curricular de Integração de Sistemas de Informação, inserida no curso de Engenharia de Sistemas Informáticos, foi desenvolvido um projeto de integração de dados.

O avanço das tecnologias de Internet das Coisas (IoT) tem vindo a transformar a forma como os dados ambientais são recolhidos, tratados e utilizados. A possibilidade de integrar sensores físicos com plataformas digitais permite o desenvolvimento de sistemas inteligentes capazes de monitorizar em tempo real variáveis como temperatura, humidade e luminosidade, promovendo decisões baseadas em dados e automatizando processos antes manuais.

Neste contexto, o presente projeto teve como objetivo a criação de um sistema completo de integração e análise de dados IoT, que demonstrasse, de forma prática, o funcionamento de um pipeline de dados, desde a recolha até à visualização dos resultados. A solução proposta combina diversas ferramentas e linguagens de programação, nomeadamente KNIME, C# e Power BI, cada uma desempenhando um papel específico dentro da arquitetura global.

O Knime foi utilizado como plataforma principal de ETL (Extract, Transform, Load), responsável pela recolha dos dados de sensores IoT e de uma API meteorológica pública (OpenWeather), pela normalização e tratamento das informações e pela sua posterior gravação em base de dados e ficheiros. Paralelamente, o módulo desenvolvido em C# permitiu o cálculo automático de estatísticas descritivas, como média, variância, desvio padrão e amplitude, e a geração de ficheiros XML complementares aos resultados obtidos. Por fim, o Power BI foi utilizado como ferramenta de visualização e análise, criando dashboards interativos que facilitam a interpretação dos dados e destacam tendências e comportamentos ambientais.

A execução do projeto foi estruturada em diferentes jobs no KNIME, assegurando modularidade e controlo sobre o processo: o ETL_IoT_API (extração, transformação e carga de dados), o LOG Manager (registo e monitorização da execução) e o Success Flow and Email Notification (validação e comunicação automática). Esta organização permitiu criar um fluxo totalmente automatizado, escalável e de fácil manutenção.

Assim, este trabalho demonstra, na prática, como a integração entre ferramentas de processamento de dados, programação e visualização analítica pode originar um sistema IoT inteligente, fiável e orientado à tomada de decisão, refletindo a importância crescente da engenharia de dados no contexto tecnológico atual.

II. Problema

O problema proposto neste projeto consistiu em desenvolver um processo completo de integração de dados provenientes de dispositivos IoT e de uma fonte meteorológica externa, demonstrando como a plataforma KNIME pode ser utilizada para extrair, transformar, combinar e carregar dados de origens distintas num sistema integrado e automatizado.

Foi simulado um cenário típico de Internet das Coisas (IoT), no qual sensores recolhem periodicamente informações ambientais — como temperatura, humidade e luminosidade — que, para serem validadas e enriquecidas, devem ser comparadas com dados de previsão meteorológica obtidos através da API pública OpenWeather. O objetivo foi, portanto, criar um processo ETL (Extract, Transform, Load) totalmente funcional que unificasse essas duas fontes de dados e disponibilizasse os resultados de forma estruturada e fiável.

O principal desafio deste trabalho residiu na integração e harmonização de dados heterogêneos, uma vez que as origens apresentavam formatos e granularidades temporais distintas. Os sensores IoT registavam medições em intervalos horários variáveis, enquanto a API fornecia previsões em blocos de três horas. Foi, portanto, necessário normalizar os formatos de data e hora, de modo a permitir o alinhamento temporal e o cruzamento correto dos registos.

2.1 Estrutura da solução

A estratégia adotada para a concretização deste objetivo passou por dividir o projeto em várias etapas, refletindo o ciclo clássico de um processo ETL. Na fase de extração, recorreu-se aos nós JSON Reader e File Reader para importar os ficheiros com dados IoT e meteorológicos. O formato JSON foi escolhido por ser o mais comum em dispositivos e APIs modernas. Para os dados IoT, foi utilizada uma simulação gerada a partir de um fluxo no **Node-RED**, representando medições horárias. Já os dados da API foram obtidos através de um nó GET Request, seguido de JSON Path para converter a resposta em colunas tabulares legíveis pelo **KNIME**.

Na etapa de transformação, foi necessário garantir que ambos os conjuntos de dados partilhavam um formato comum, especialmente no campo da data e hora. Este ponto revelou-se um dos principais desafios do projeto, uma vez que as leituras IoT apresentavam segundos e minutos variáveis, enquanto a API devolvia blocos de tempo fixos (por exemplo, de 3 em 3 horas). Para resolver este problema, foi utilizada a função String Manipulation, onde as datas foram formatadas no padrão (yyyy-MM-dd HH:mm:ss) e arredondadas para múltiplos de três horas. Este processo permitiu alinhar temporalmente as duas fontes, tornando possível o cruzamento direto entre elas.

Depois de uniformizados os dados, a junção foi efetuada com o nó Joiner, tendo como chave o campo de data/hora formatado. Este passo permitiu combinar, para cada registo IoT, a respetiva informação meteorológica, criando um dataset enriquecido. A partir daí, aplicaram-se transformações adicionais com os nós Math Formula e GroupBy, para calcular médias, desvios e outras medidas de interesse, representando indicadores de qualidade das medições recolhidas.

Concluída a transformação, iniciou-se a fase de carregamento (Load). O conjunto de dados final foi enviado para uma base de dados **MySQL** através do nó DB Connector (ligado ao servidor local) e DB Writer, criando uma tabela consolidada chamada dados_iot_api. Em paralelo, o mesmo resultado foi exportado em dois formatos: **CSV**, para permitir a leitura no Excel ou Power BI, e **XML**, para armazenamento e consulta estruturada em sistemas externos. Estes ficheiros foram guardados com nomes dinâmicos, que incluíam a data de execução, como `resumo_diario_2025-10-18.csv`, garantindo assim uma organização temporal dos resultados.

Uma parte importante do projeto consistiu na implementação, em **C#**, de um módulo responsável pelo **cálculo estatístico das variáveis recolhidas** durante o processo ETL. Apesar de o KNIME permitir realizar operações matemáticas com os nós **Math Formula** e **GroupBy**, optou-se por desenvolver o processamento estatístico em **C#** para demonstrar a integração entre diferentes tecnologias e linguagens, bem como para tirar partido da eficiência da linguagem no tratamento de grandes volumes de dados.

O código em **C#** foi desenvolvido numa aplicação de consola simples (**Visual Studio Installer**), que recebe como entrada um ficheiro **CSV gerado pelo KNIME**, contendo as leituras diárias de temperatura, humidade, temperatura máxima/mínima. A aplicação lê os dados, processa as colunas e calcula medidas estatísticas como **média**, **moda**, **variância**, **desvio padrão** e **amplitude**, produzindo como resultado um novo ficheiro XML.

2.2 Visão Geral do Funcionamento

O sistema segue um fluxo **ETL (Extract, Transform, Load)** composto por etapas encadeadas:

- **Extração:**
 - Leitura dos dados IoT (JSON Reader).
 - Recolha de dados meteorológicos via API OpenWeather (GET Request + JSON Path).
- **Transformação:**
 - Limpeza e normalização de datas e formatos (String Manipulation, Rule Engine).
 - Junção dos datasets (Joiner).
 - Cálculo de médias e indicadores (Math Formula, GroupBy).
 - Cálculos estatísticos adicionais em **C#** (média, variância, desvio padrão, amplitude).
- **Carregamento:**
 - Armazenamento no **MySQL** (DB Writer).
 - Exportação em **CSV** e **XML** com nomes automáticos.
- **Controlo e Logs**
 - Registo automático do estado de execução.
 - Criação de variáveis com mensagens e datas (Java Edit Variable).
 - Gravação dos logs em **CSV** com data e hora automáticas (CSV Writer).

O resultado é um **processo automatizado, robusto e reutilizável**, capaz de integrar e monitorizar dados de diferentes origens com total rastreabilidade.

III. Estratégia Utilizada

A execução do projeto seguiu o modelo clássico **ETL**, dividido em fases bem definidas: extração, transformação e carregamento.

Durante a **extração**, foram utilizados os nós JSON Reader e File Reader para importar os ficheiros IoT e meteorológicos. O nó GET Request, associado ao JSON Path, foi responsável por recolher os dados da API e converter a resposta JSON em colunas tabulares.

Na fase de **transformação**, aplicaram-se nós de manipulação de texto e lógica, como String Manipulation, Math Formula, Rule Engine e GroupBy, com o objetivo de:

- Uniformizar o formato das datas (yyyy-MM-dd HH:mm:ss);
- Arredondar horas para múltiplos de 3h (para coincidir com a API);
- Calcular médias e indicadores estatísticos;
- Eliminar colunas redundantes e corrigir dados nulos.

Finalmente, na fase de **carregamento**, utilizou-se o **DB Connector** e o **DB Writer** para guardar os resultados na base de dados MySQL. Os mesmos dados foram ainda exportados em **CSV** e **XML** com nomes automáticos contendo a data de execução.

3.1 Extração

A fase de extração consistiu na recolha automática de informações a partir de duas fontes principais:

1. **Sensores IoT** — responsáveis por gerar dados ambientais locais, como:
 - Temperatura
 - Humidade
 - LuminosidadeEstes dados eram registados com data e hora, simulando um ambiente real de monitorização contínua.
2. **API Meteorológica (OpenWeather)** — utilizada para recolher previsões de temperatura e condições atmosféricas externas, servindo como base de comparação com os sensores.

O processo foi implementado no **Knime**, com ligações automáticas que:

- Fazem *fetch* dos dados IoT e da API em intervalos definidos;
- Guardam os resultados em ficheiros **CSV** e também em uma base de dados **MySQL**;
- Asseguram a **coerência temporal** dos registos (datas e horas normalizadas).

Aplicação do Processo ETL

Essa automação permite que, sempre que novos dados são recolhidos, o fluxo ETL seja executado novamente e os resultados sejam atualizados no Power BI, refletindo a informação mais recente disponível.

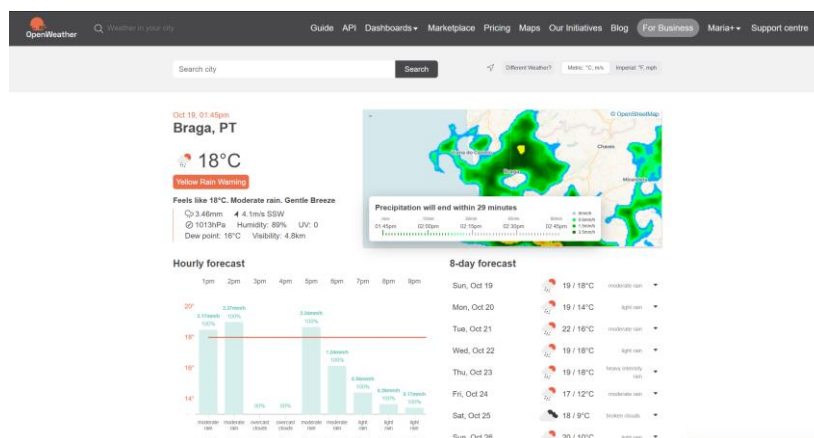


Figura 1 – OpenWeather API

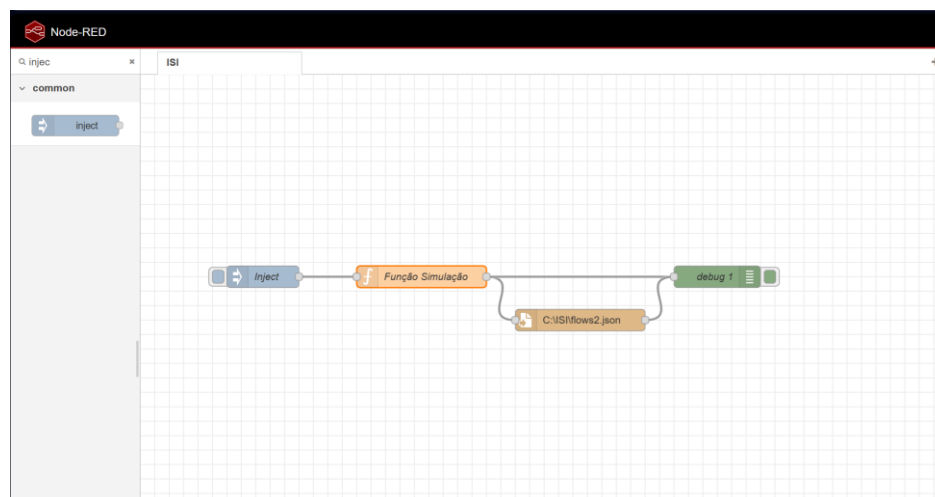


Figura 2 - Fluxo Node-RED

Aplicação do Processo ETL



ESCOLA
SUPERIOR
DE TECNOLOGIA

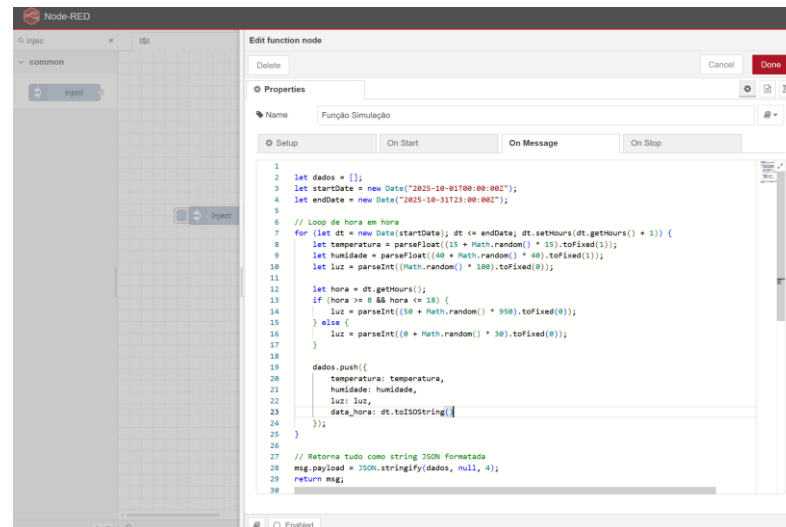


Figura 3 - Lógica Node-RED

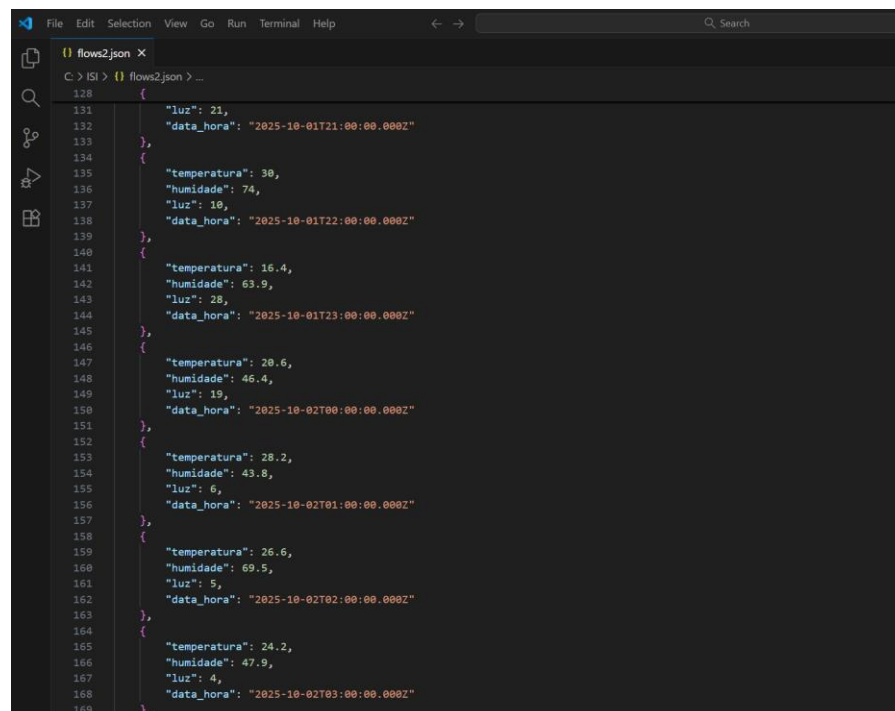


Figura 4 - Dados Simulados

Aplicação do Processo ETL

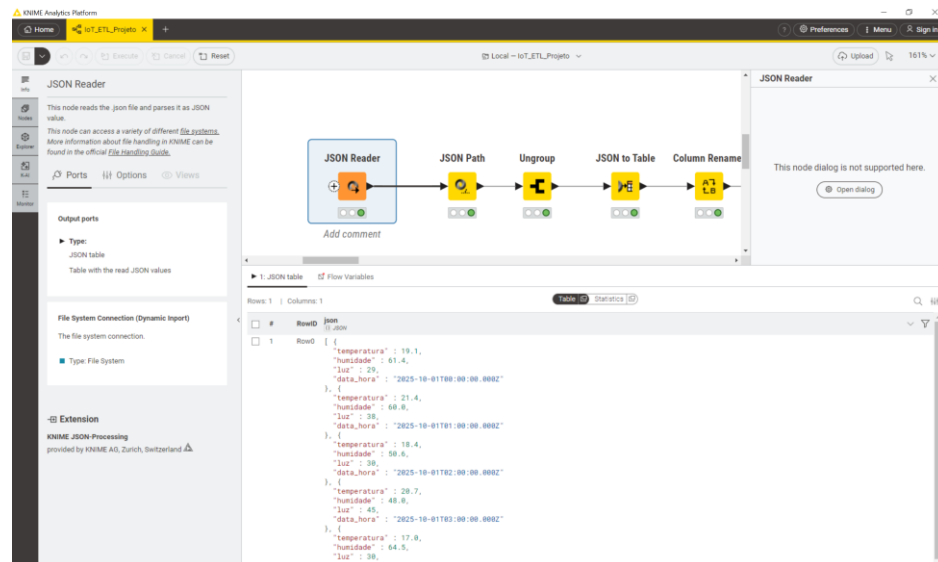


Figura 5 - Leitura Dados Simulados

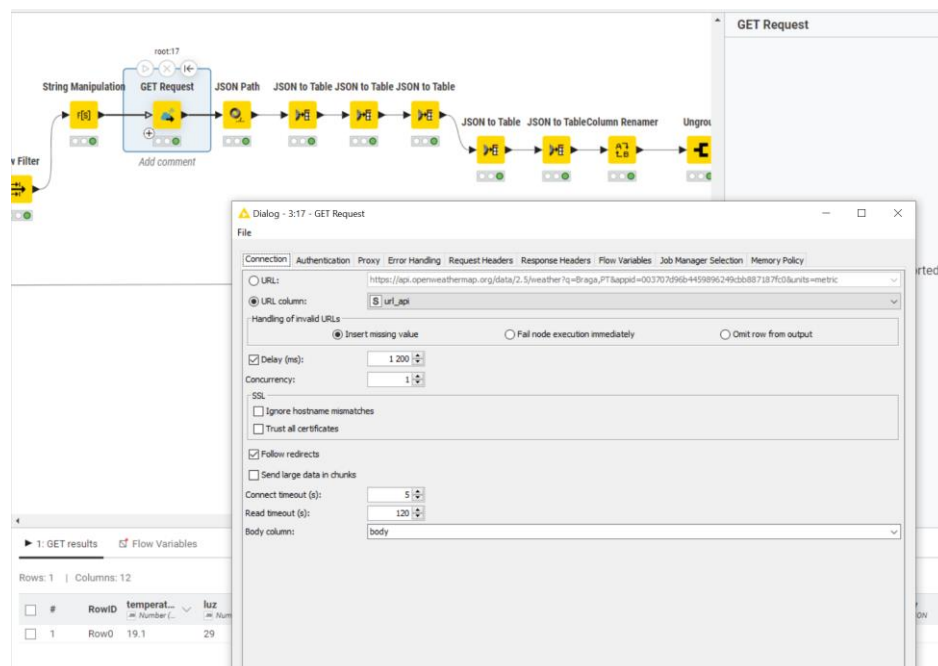


Figura 6 - Leitura Dados API

3.2 Transformação

Nesta fase, os dados extraídos foram limpos, combinados e padronizados para assegurar consistência e integridade.

As principais transformações incluíram:

- **Conversão de formatos de data/hora** para padrão ISO (yyyy-MM-dd HH:mm:ss);
- **Tratamento de valores nulos e duplicados;**
- **Normalização das unidades de medida;**
- **Junção** dos datasets de IoT e API através da data/hora comum;
- **Criação de colunas derivadas**, como médias diárias e variações entre sensores e API;
- **Geração de logs automáticos** de execução e estatísticas do processo.

Essas operações foram totalmente automatizadas dentro do KNIME, permitindo executar o fluxo completo sem intervenção manual.

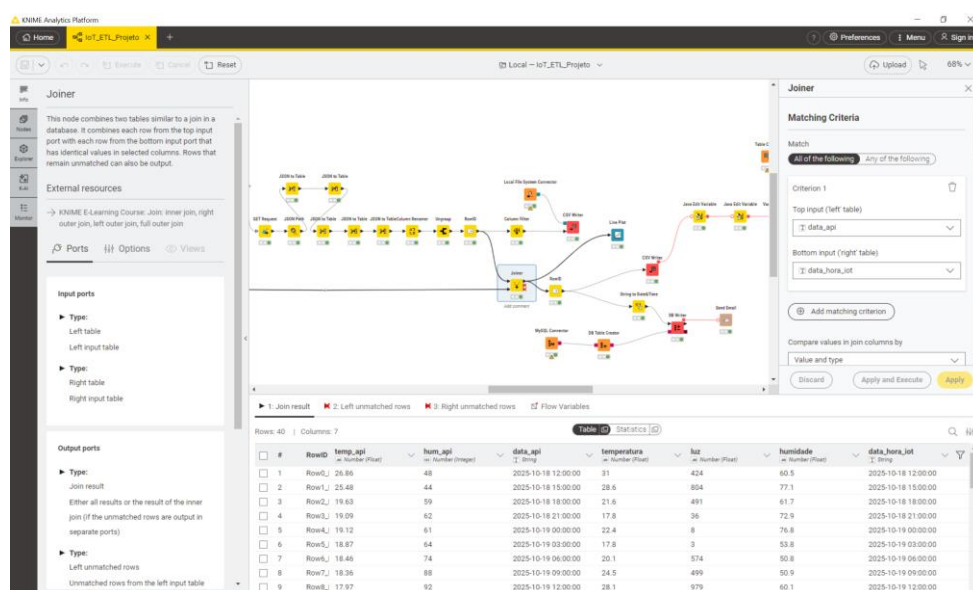


Figura 7 - Joiner

Aplicação do Processo ETL

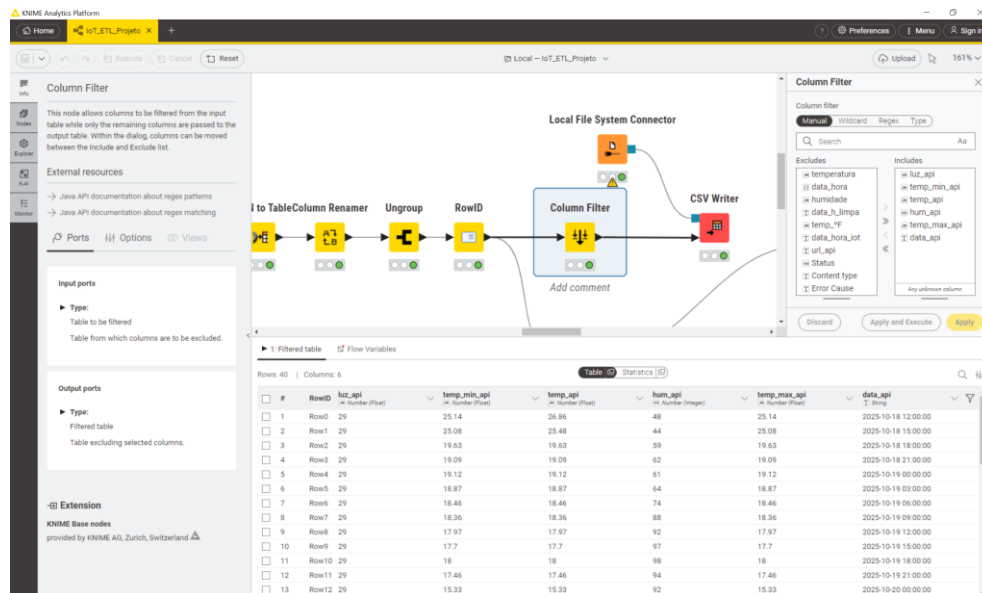


Figura 8 - Dados API Transformados

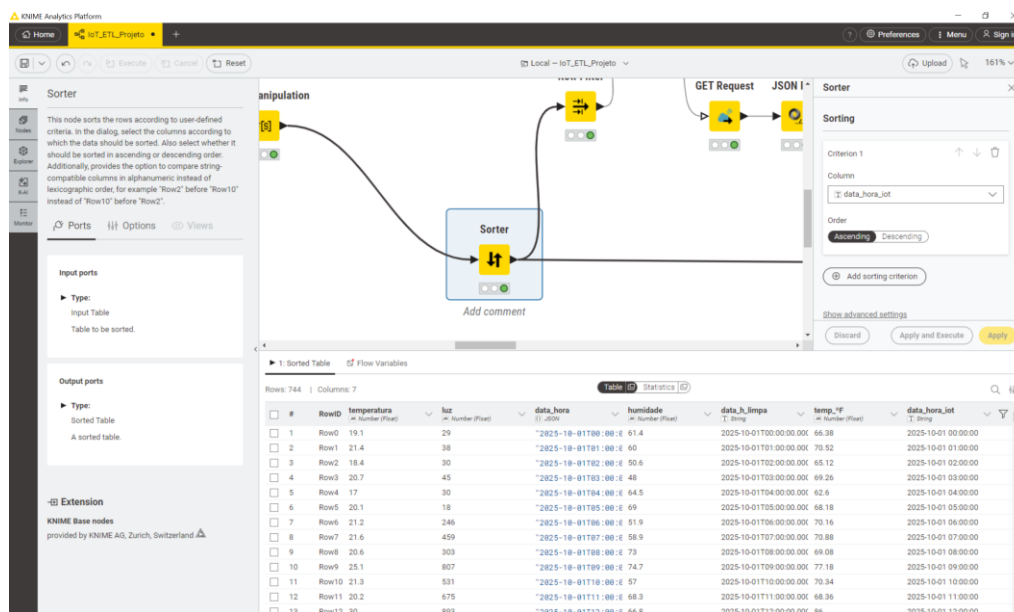


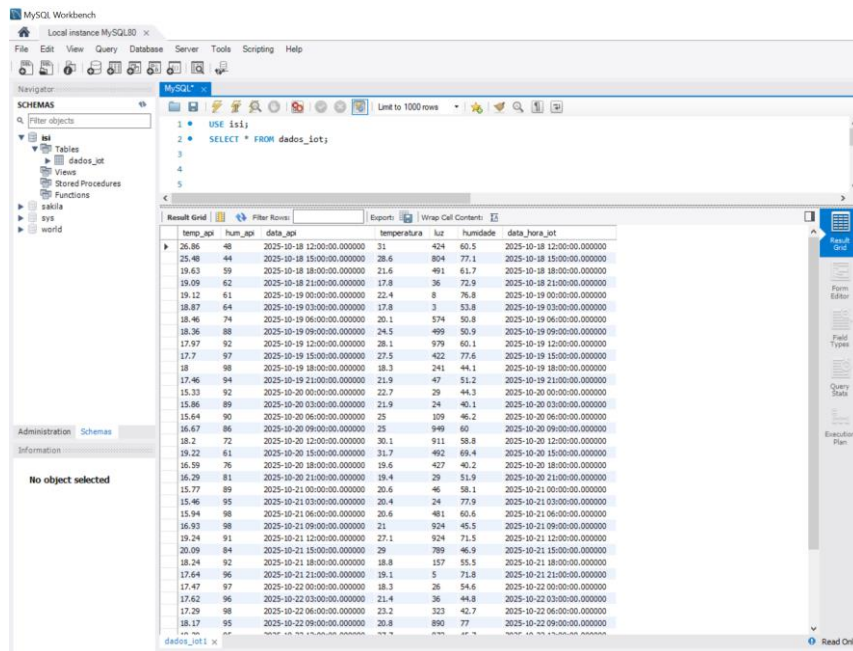
Figura 9 - Dados IoT Transformados

3.3 Carregamento

Após a transformação, os dados tratados foram armazenados e disponibilizados para análise em múltiplos formatos e plataformas:

- **Base de Dados MySQL** - os registos finais são gravados na tabela dados_iot, garantindo persistência e permitindo consultas SQL diretas.
- **Ficheiros CSV** - exportados automaticamente pelo KNIME, servindo como cópia local e suporte para integração com outras ferramentas.
- **Ficheiros XML** - gerados através de uma aplicação desenvolvida em C#, que processa os dados mais recentes e inclui estatísticas agregadas (como médias, máximos, mínimos e variações diárias).
- **Dashboards Interativos (Power BI)** - conectados à base MySQL, exibindo de forma dinâmica as métricas de temperatura, humidade e tendências temporais.

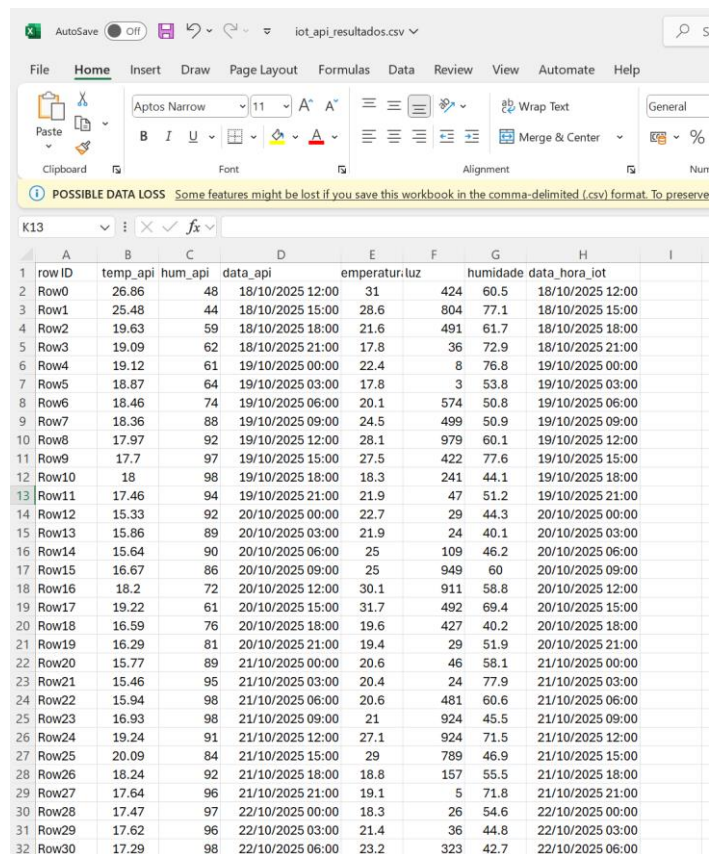
Graças à automação do fluxo no Knime e ao módulo em C#, todo o sistema é atualizado de forma sincronizada: sempre que novos dados são recolhidos, as tabelas, ficheiros e visualizações refletem automaticamente a informação mais recente.



temp_apt	hum_apt	data_apt	temperatura	luz	humidade	data_hora_apt
26.96	40	2025-10-18 12:00:00.000000	31	424	60.5	2025-10-18 12:00:00.000000
25.48	44	2025-10-18 15:00:00.000000	28.6	804	77.1	2025-10-18 15:00:00.000000
19.63	59	2025-10-18 18:00:00.000000	21.6	491	61.7	2025-10-18 18:00:00.000000
19.09	62	2025-10-18 21:00:00.000000	17.8	36	72.9	2025-10-18 21:00:00.000000
19.12	61	2025-10-19 00:00:00.000000	22.4	8	76.8	2025-10-19 00:00:00.000000
18.87	64	2025-10-19 03:00:00.000000	17.8	3	53.8	2025-10-19 03:00:00.000000
18.46	74	2025-10-19 06:00:00.000000	20.1	574	50.8	2025-10-19 06:00:00.000000
18.36	88	2025-10-19 09:00:00.000000	24.5	499	50.9	2025-10-19 09:00:00.000000
17.97	92	2025-10-19 12:00:00.000000	28.1	979	60.1	2025-10-19 12:00:00.000000
17.7	97	2025-10-19 15:00:00.000000	27.5	422	77.6	2025-10-19 15:00:00.000000
18	98	2025-10-19 18:00:00.000000	18.3	241	44.1	2025-10-19 18:00:00.000000
17.46	94	2025-10-19 21:00:00.000000	21.9	47	51.2	2025-10-19 21:00:00.000000
15.33	92	2025-10-20 00:00:00.000000	22.7	29	44.3	2025-10-20 00:00:00.000000
15.86	89	2025-10-20 03:00:00.000000	21.9	24	40.1	2025-10-20 03:00:00.000000
15.64	90	2025-10-20 06:00:00.000000	25	109	46.2	2025-10-20 06:00:00.000000
16.67	86	2025-10-20 09:00:00.000000	25	949	60	2025-10-20 09:00:00.000000
18.2	72	2025-10-20 12:00:00.000000	30.1	911	58.8	2025-10-20 12:00:00.000000
19.22	61	2025-10-20 15:00:00.000000	31.7	462	69.4	2025-10-20 15:00:00.000000
16.59	76	2025-10-20 18:00:00.000000	19.6	427	40.2	2025-10-20 18:00:00.000000
16.29	81	2025-10-20 21:00:00.000000	19.4	29	51.9	2025-10-20 21:00:00.000000
15.77	89	2025-10-21 00:00:00.000000	20.6	46	58.1	2025-10-21 00:00:00.000000
15.46	95	2025-10-21 03:00:00.000000	20.4	24	77.9	2025-10-21 03:00:00.000000
15.94	98	2025-10-21 06:00:00.000000	20.6	481	60.6	2025-10-21 06:00:00.000000
16.93	98	2025-10-21 09:00:00.000000	21	924	45.5	2025-10-21 09:00:00.000000
19.24	91	2025-10-21 12:00:00.000000	27.1	924	71.5	2025-10-21 12:00:00.000000
20.09	94	2025-10-21 15:00:00.000000	29	789	46.9	2025-10-21 15:00:00.000000
18.24	92	2025-10-21 18:00:00.000000	18.8	157	55.5	2025-10-21 18:00:00.000000
17.64	96	2025-10-21 21:00:00.000000	19.1	5	71.8	2025-10-21 21:00:00.000000
17.47	97	2025-10-22 00:00:00.000000	18.3	36	54.6	2025-10-22 00:00:00.000000
17.62	96	2025-10-22 03:00:00.000000	21.4	38	44.8	2025-10-22 03:00:00.000000
17.29	98	2025-10-22 06:00:00.000000	23.2	323	42.7	2025-10-22 06:00:00.000000
18.17	95	2025-10-22 09:00:00.000000	20.8	890	77	2025-10-22 09:00:00.000000

Figura 10 - Dados MySQL

Aplicação do Processo ETL



AutoSave Off iot_api_resultados.csv

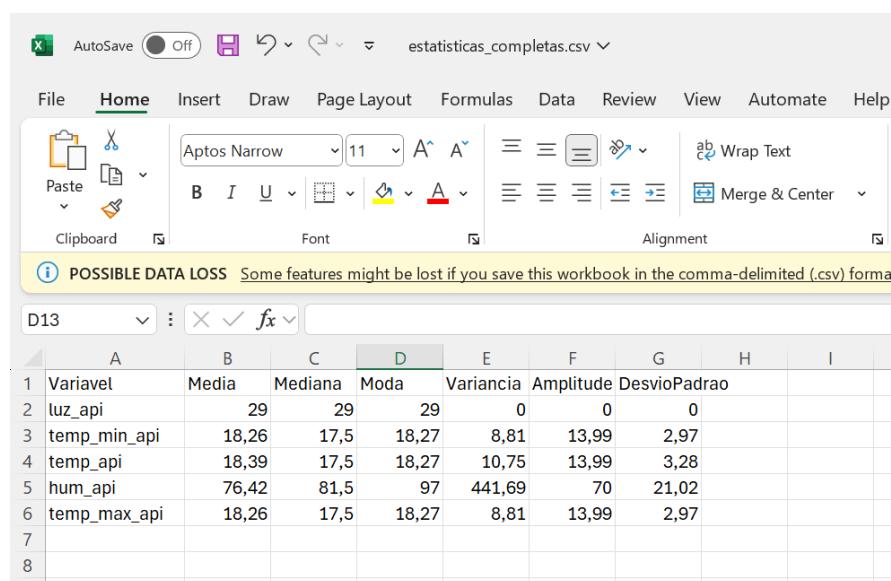
File Home Insert Draw Page Layout Formulas Data Review View Automate Help

Paste Clipboard Font Alignment

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve

	A	B	C	D	E	F	G	H	I
1	row ID	temp_api	hum_api	data_api	emperatur:luz	humidade	data_hora_iot		
2	Row0	26.86	48	18/10/2025 12:00	31	424	60.5	18/10/2025 12:00	
3	Row1	25.48	44	18/10/2025 15:00	28.6	804	77.1	18/10/2025 15:00	
4	Row2	19.63	59	18/10/2025 18:00	21.6	491	61.7	18/10/2025 18:00	
5	Row3	19.09	62	18/10/2025 21:00	17.8	36	72.9	18/10/2025 21:00	
6	Row4	19.12	61	19/10/2025 00:00	22.4	8	76.8	19/10/2025 00:00	
7	Row5	18.87	64	19/10/2025 03:00	17.8	3	53.8	19/10/2025 03:00	
8	Row6	18.46	74	19/10/2025 06:00	20.1	574	50.8	19/10/2025 06:00	
9	Row7	18.36	88	19/10/2025 09:00	24.5	499	50.9	19/10/2025 09:00	
10	Row8	17.97	92	19/10/2025 12:00	28.1	979	60.1	19/10/2025 12:00	
11	Row9	17.7	97	19/10/2025 15:00	27.5	422	77.6	19/10/2025 15:00	
12	Row10	18	98	19/10/2025 18:00	18.3	241	44.1	19/10/2025 18:00	
13	Row11	17.46	94	19/10/2025 21:00	21.9	47	51.2	19/10/2025 21:00	
14	Row12	15.33	92	20/10/2025 00:00	22.7	29	44.3	20/10/2025 00:00	
15	Row13	15.86	89	20/10/2025 03:00	21.9	24	40.1	20/10/2025 03:00	
16	Row14	15.64	90	20/10/2025 06:00	25	109	46.2	20/10/2025 06:00	
17	Row15	16.67	86	20/10/2025 09:00	25	949	60	20/10/2025 09:00	
18	Row16	18.2	72	20/10/2025 12:00	30.1	911	58.8	20/10/2025 12:00	
19	Row17	19.22	61	20/10/2025 15:00	31.7	492	69.4	20/10/2025 15:00	
20	Row18	16.59	76	20/10/2025 18:00	19.6	427	40.2	20/10/2025 18:00	
21	Row19	16.29	81	20/10/2025 21:00	19.4	29	51.9	20/10/2025 21:00	
22	Row20	15.77	89	21/10/2025 00:00	20.6	46	58.1	21/10/2025 00:00	
23	Row21	15.46	95	21/10/2025 03:00	20.4	24	77.9	21/10/2025 03:00	
24	Row22	15.94	98	21/10/2025 06:00	20.6	481	60.6	21/10/2025 06:00	
25	Row23	16.93	98	21/10/2025 09:00	21	924	45.5	21/10/2025 09:00	
26	Row24	19.24	91	21/10/2025 12:00	27.1	924	71.5	21/10/2025 12:00	
27	Row25	20.09	84	21/10/2025 15:00	29	789	46.9	21/10/2025 15:00	
28	Row26	18.24	92	21/10/2025 18:00	18.8	157	55.5	21/10/2025 18:00	
29	Row27	17.64	96	21/10/2025 21:00	19.1	5	71.8	21/10/2025 21:00	
30	Row28	17.47	97	22/10/2025 00:00	18.3	26	54.6	22/10/2025 00:00	
31	Row29	17.62	96	22/10/2025 03:00	21.4	36	44.8	22/10/2025 03:00	
32	Row30	17.29	98	22/10/2025 06:00	23.2	323	42.7	22/10/2025 06:00	

Figura 11 - Dados CSV Joiner



AutoSave Off estatisticas_completas.csv

File Home Insert Draw Page Layout Formulas Data Review View Automate Help

Paste Clipboard Font Alignment

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format.

	A	B	C	D	E	F	G	H	I
1	Variavel	Media	Mediana	Moda	Variancia	Amplitude	DesvioPadrao		
2	luz_api	29	29	29	0	0	0		
3	temp_min_api	18,26	17,5	18,27	8,81	13,99	2,97		
4	temp_api	18,39	17,5	18,27	10,75	13,99	3,28		
5	hum_api	76,42	81,5	97	441,69	70	21,02		
6	temp_max_api	18,26	17,5	18,27	8,81	13,99	2,97		
7									
8									

Figura 12 - Dados CSV das Estatísticas geradas em C#

Aplicação do Processo ETL



ESCOLA
SUPERIOR
DE TECNOLOGIA

AutoSave Off api_resultados.csv

File Home Insert Draw Page Layout Formulas Data Review View

Paste Clipboard Font Alignment

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the common format.

111

	A	B	C	D	E	F	G
1	row ID	temp_min	temp_api	hum_api	temp_max	data_api	
2	Row0	25.14	26.86	48	25.14	18/10/2025 12:00	
3	Row1	25.08	25.48	44	25.08	18/10/2025 15:00	
4	Row2	19.63	19.63	59	19.63	18/10/2025 18:00	
5	Row3	19.09	19.09	62	19.09	18/10/2025 21:00	
6	Row4	19.12	19.12	61	19.12	19/10/2025 00:00	
7	Row5	18.87	18.87	64	18.87	19/10/2025 03:00	
8	Row6	18.46	18.46	74	18.46	19/10/2025 06:00	
9	Row7	18.36	18.36	88	18.36	19/10/2025 09:00	
10	Row8	17.97	17.97	92	17.97	19/10/2025 12:00	
11	Row9	17.7	17.7	97	17.7	19/10/2025 15:00	
12	Row10	18	18	98	18	19/10/2025 18:00	
13	Row11	17.46	17.46	94	17.46	19/10/2025 21:00	
14	Row12	15.33	15.33	92	15.33	20/10/2025 00:00	
15	Row13	15.86	15.86	89	15.86	20/10/2025 03:00	
16	Row14	15.64	15.64	90	15.64	20/10/2025 06:00	
17	Row15	16.67	16.67	86	16.67	20/10/2025 09:00	
18	Row16	18.2	18.2	72	18.2	20/10/2025 12:00	
19	Row17	19.22	19.22	61	19.22	20/10/2025 15:00	
20	Row18	16.59	16.59	76	16.59	20/10/2025 18:00	
21	Row19	16.29	16.29	81	16.29	20/10/2025 21:00	
22	Row20	15.77	15.77	89	15.77	21/10/2025 00:00	
23	Row21	15.46	15.46	95	15.46	21/10/2025 03:00	
24	Row22	15.94	15.94	98	15.94	21/10/2025 06:00	
25	Row23	16.93	16.93	98	16.93	21/10/2025 09:00	
26	Row24	19.24	19.24	91	19.24	21/10/2025 12:00	
27	Row25	20.09	20.09	84	20.09	21/10/2025 15:00	
28	Row26	18.24	18.24	92	18.24	21/10/2025 18:00	
29	Row27	17.64	17.64	96	17.64	21/10/2025 21:00	
30	Row28	17.47	17.47	97	17.47	22/10/2025 00:00	

Figura 13 - Dados CSV API

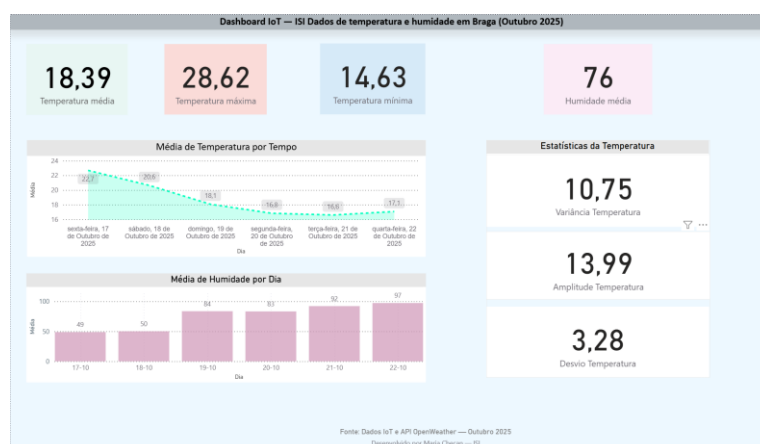
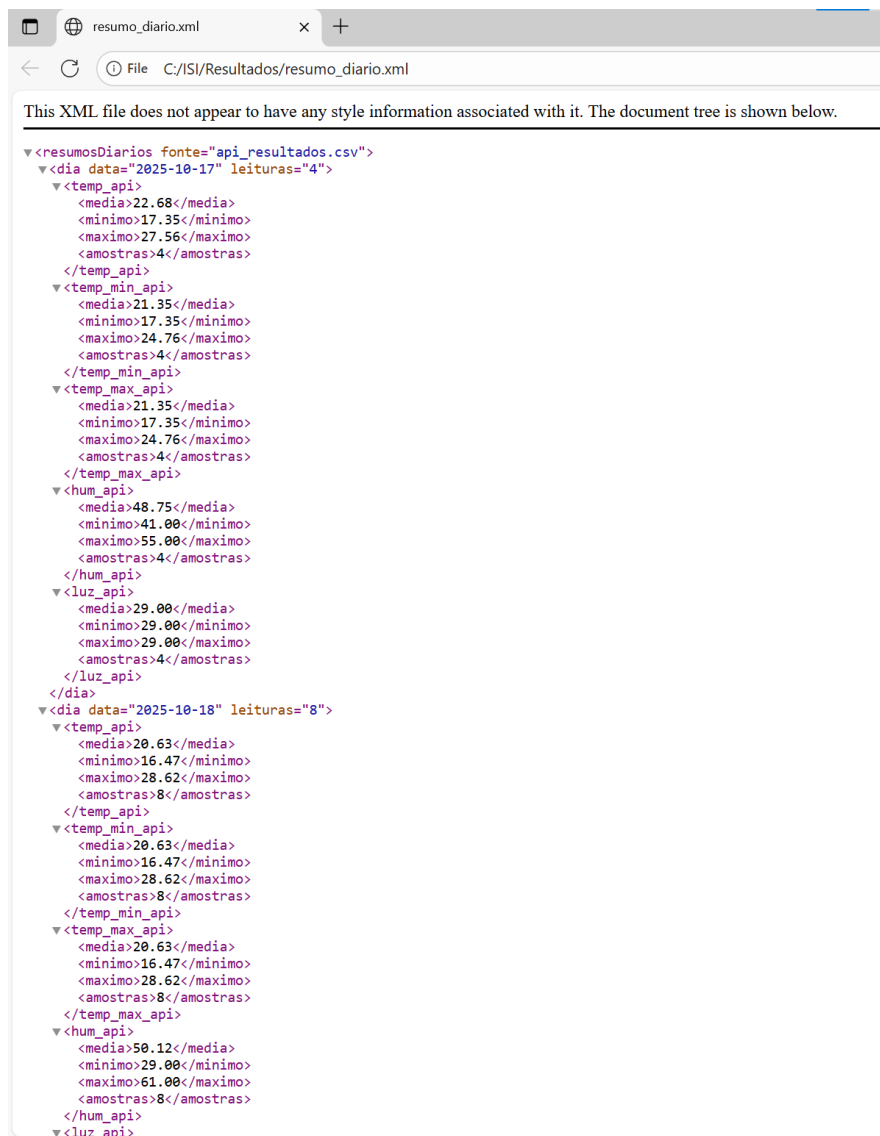


Figura 14 - PowerBI



```
<?xml version="1.0" encoding="utf-8" />
<resumosDiarios fonte="api_resultados.csv">
  <dia data="2025-10-17" leituras="4">
    <temp_api>
      <media>22.68</media>
      <minimo>17.35</minimo>
      <maximo>27.56</maximo>
      <amostras>4</amostras>
    </temp_api>
    <temp_min_api>
      <media>21.35</media>
      <minimo>17.35</minimo>
      <maximo>24.76</maximo>
      <amostras>4</amostras>
    </temp_min_api>
    <temp_max_api>
      <media>21.35</media>
      <minimo>17.35</minimo>
      <maximo>24.76</maximo>
      <amostras>4</amostras>
    </temp_max_api>
    <hum_api>
      <media>48.75</media>
      <minimo>41.00</minimo>
      <maximo>55.00</maximo>
      <amostras>4</amostras>
    </hum_api>
    <luz_api>
      <media>29.00</media>
      <minimo>29.00</minimo>
      <maximo>29.00</maximo>
      <amostras>4</amostras>
    </luz_api>
  </dia>
  <dia data="2025-10-18" leituras="8">
    <temp_api>
      <media>20.63</media>
      <minimo>16.47</minimo>
      <maximo>28.62</maximo>
      <amostras>8</amostras>
    </temp_api>
    <temp_min_api>
      <media>20.63</media>
      <minimo>16.47</minimo>
      <maximo>28.62</maximo>
      <amostras>8</amostras>
    </temp_min_api>
    <temp_max_api>
      <media>20.63</media>
      <minimo>16.47</minimo>
      <maximo>28.62</maximo>
      <amostras>8</amostras>
    </temp_max_api>
    <hum_api>
      <media>50.12</media>
      <minimo>29.00</minimo>
      <maximo>61.00</maximo>
      <amostras>8</amostras>
    </hum_api>
    <luz_api>
```

Figura 15 - Dados XML gerado em C#

IV. Jobs

4.1 Job 1 — ETL_IoT_API

O Job 1 corresponde ao processo principal de integração e tratamento de dados. Nele, foi implementado todo o fluxo ETL (Extract, Transform, Load) responsável por unir as fontes IoT e API Meteorológica (OpenWeather).

As suas principais funções incluem:

- Extração de dados dos sensores IoT e da API pública, garantindo a leitura automática e periódica das fontes;
- Transformação e limpeza dos dados, com normalização de formatos, tratamento de nulos e unificação de unidades;
- Junção (Join) entre os datasets de IoT e API através de data/hora, permitindo comparações diretas;
- Cálculo de médias e agregações por período (dia/hora) e exportação dos resultados;
- Carga (Load) final em múltiplos destinos:
 - Base de dados MySQL, para armazenamento persistente;
 - Ficheiro CSV, como backup e integração com Power BI;
 - Ficheiro XML, gerado externamente pelo módulo em C#, contendo estatísticas complementares.

O Job ETL_IoT_API é o núcleo do sistema, garantindo que toda a cadeia de dados, desde a recolha até à disponibilização.

4.2 Job 2 — LOG Manager

O Job 2 foi desenvolvido com o objetivo de gerir e monitorizar a execução dos processos, assegurando o controlo e a rastreabilidade do ETL.

As suas principais responsabilidades são:

- Registo automático de logs de execução (início, fim e estado de cada componente);
- Armazenamento dos resultados de logs em ficheiros CSV e/ou na base de dados MySQL;
- Geração de relatórios de execução, facilitando a validação do pipeline de dados.

O Job LOG Manager atua como camada de controlo do sistema, permitindo uma visão clara do desempenho e garantindo a confiabilidade e transparência das operações do projeto.

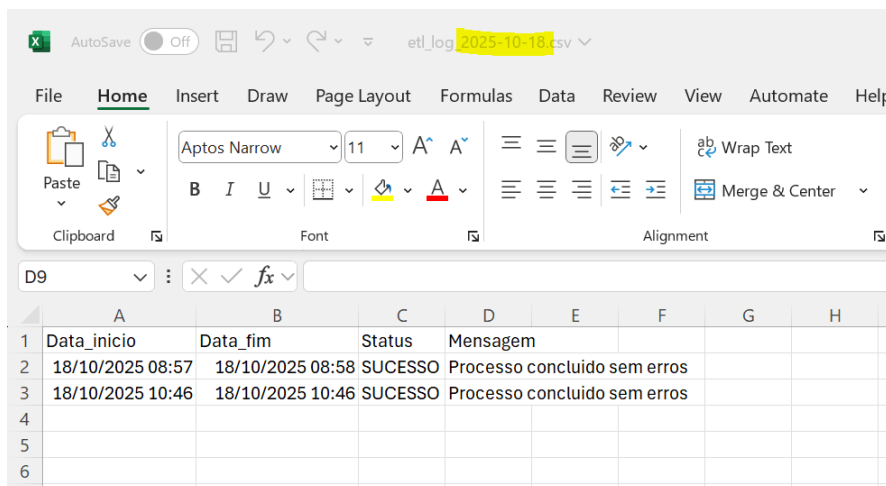
4.3 Job 3 — Success Flow and Email Notification

O Job 3 foi criado para encerrar o processo ETL de forma automatizada e controlada, garantindo a comunicação imediata sobre o estado da execução. Ele é acionado após a conclusão bem-sucedida dos *jobs* anteriores (ETL_IoT_API e LOG Manager), funcionando como uma camada de monitorização e notificação.

As suas principais funções são:

- Receção do sinal de sucesso (*success flow*) gerado no final do *Job ETL_IoT_API*, confirmando que todas as etapas — extração, transformação e carga — foram executadas sem erros;
- Validação de integridade dos ficheiros e registos finais (CSV, XML e base MySQL), assegurando que todos os outputs esperados foram produzidos;
- Envio automático de e-mail de notificação, através de um conector de e-mail configurado no KNIME.

O Success Flow and Email Notification representa a última etapa do ciclo automatizado de dados, fechando o pipeline com notificações em tempo real e garantia de execução completa, tornando o sistema autossuficiente, auditável e confiável.



	A	B	C	D	E	F	G	H
1	Data_inicio	Data_fim	Status	Mensagem				
2	18/10/2025 08:57	18/10/2025 08:58	SUCESSO	Processo concluído sem erros				
3	18/10/2025 10:46	18/10/2025 10:46	SUCESSO	Processo concluído sem erros				
4								
5								
6								

Figura 16 - Ficheiro CSV dos Logs

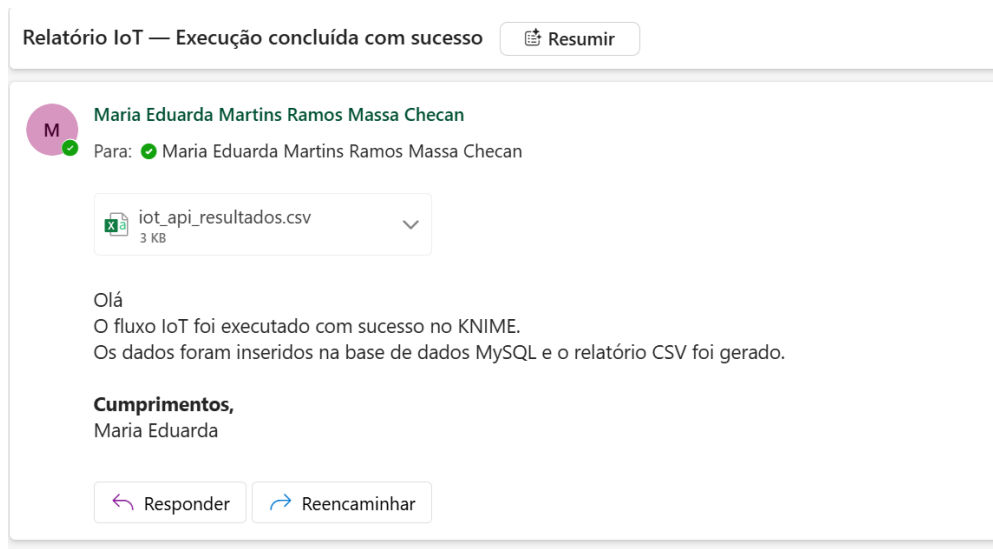


Figura 17 - Notificação de Email após conclusão do fluxo ETL

V. Vídeo com demonstração (QR Code)

Escaneie o QR abaixo para visualizar o vídeo de demonstração do projeto:



Figura 18 - QR Code

VI. Conclusão e Trabalhos Futuros

O presente projeto demonstrou a implementação de um processo completo de integração e análise de dados IoT, englobando todas as fases de um pipeline de dados, desde a extração até à visualização analítica. Através da utilização conjunta do KNIME, C# e Power BI, foi possível desenvolver uma solução automatizada, modular e fiável, capaz de recolher, transformar, armazenar e apresentar dados ambientais de forma estruturada e eficiente.

O Job ETL_IoT_API desempenhou um papel central neste processo, assegurando a extração simultânea dos dados provenientes dos sensores IoT e da API meteorológica pública (OpenWeather), a sua transformação e normalização, bem como o carregamento dos resultados em múltiplos formatos. Já o LOG Manager foi responsável pelo controlo de execução e pela criação de registos detalhados das operações, aumentando a transparência e a rastreabilidade do sistema. Por sua vez, o Success Flow and Email Notification encerrou o processo de forma automatizada, validando a integridade dos ficheiros e enviando notificações automáticas de sucesso por e-mail.

Complementarmente, a integração com C# permitiu gerar ficheiros XML enriquecidos com estatísticas descritivas, como médias, máximos, mínimos, desvio padrão e amplitude térmica. Estes resultados foram posteriormente integrados com o Power BI, que possibilitou a criação de dashboards interativos, traduzindo os dados em informação visual e intuitiva, facilitando a interpretação e a análise das tendências ambientais registadas.

De forma geral, o sistema atingiu os objetivos propostos, garantindo a integração automática das fontes de dados, a consistência dos registos, o armazenamento estruturado e a visualização dinâmica das informações. O projeto demonstrou, assim, a capacidade de articular diferentes tecnologias num fluxo de dados coerente e totalmente automatizado.

Como trabalhos futuros, propõe-se a expansão do sistema para incluir novos tipos de sensores e variáveis ambientais, bem como a implementação de modelos preditivos no KNIME, capazes de antecipar variações de temperatura e humidade. Além disso, pretende-se incorporar alertas inteligentes por e-mail ou notificações push em caso de anomalias nos dados, publicar o dashboard Power BI online com atualizações automáticas e migrar a infraestrutura para uma plataforma em nuvem, de modo a garantir maior escalabilidade e desempenho.

Com estas melhorias, o projeto poderá evoluir para uma plataforma IoT completa, inteligente e preditiva, apta a apoiar decisões baseadas em dados em tempo real e a contribuir para a modernização de processos de monitorização ambiental.

VII. Referências Bibliográficas

- KNIME Documentation: <https://docs.knime.com>
- KNIME Hub (exemplos de ETL): <https://hub.knime.com/>
- MySQL Docs: <https://dev.mysql.com/doc/>
- OpenWeather API: <https://openweathermap.org/api>
- Stack Overflow — fóruns de integração KNIME e SQL
- YouTube — tutoriais “KNIME ETL Project”
- Material do docente e exemplos fornecidos em aula