IPCA INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

ANÁLISE DE DADOS DE TELEMETRIA - CAUDAIS

INTEGRAÇÃO DE SISTEMAS DE INFORMAÇÃO

Aluno:

João Filipe Ferreira (a25275)

Docente:

Luís Ferreira

ENGENHARIA DE SISTEMAS INFORMÁTICOS

Outubro, 2025

IPCA INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

ANÁLISE DE DADOS DE TELEMETRIA – CAUDAIS

INTEGRAÇÃO DE SISTEMAS DE INFORMAÇÃO

Aluno:

João Filipe Ferreira (a25275)

Docente:

Luís Ferreira

ENGENHARIA DE SISTEMAS INFORMÁTICOS

Outubro, 2025

Índice

Índice de figuras

Problema

Este projeto apresenta o desenvolvimento de um Pipeline de ETL para normalização e enriquecimento de dados de telemetria de consumos de caudal.

O projeto visa resolver uma necessidade de processar dados brutos de telemetria de caudal de múltiplos sensores, que são fornecidos (via FTP) em ficheiros CSV desnormalizados e inconsistentes, e enriquecê-los com dados externos (Precipitação) para relacionamento e análise de contexto.

Fontes de Dados: Dados de telemetria de consumos de caudal (CSV), com periodicidade de 15 minutos e API Externa de Precipitação (JSON).

Objetivos Específicos:

- ✓ Implementar a lógica de cálculo do Consumo Diário (diferença entre leituras das 00 horas de cada dia).
- ✓ Demonstrar a Integração de Sistemas através do consumo de uma API (JSON).
- ✓ Implementar Orquestração de processos para automação e load por sensor.
- ✓ Superar e documentar as limitações de compatibilidade do software KNIME.

Este projeto demonstra a aplicabilidade prática das ferramentas ETL na área da análise de dados, evidenciando a importância da automatização e da integração de fontes distintas para a produção de informação consolidada e de valor analítico.

Estratégia utilizada

O projeto recorreu a diversas etapas de transformação no KNIME Analytics Platform, através de uma estratégia de Pipeline (ETL).

- Na fase Extract (E), foi implementado um Acesso Remoto (FTP/SSH) e Orquestração de Processos para transferência dos ficheiros CSV. Uma orquestração via loop (Loop Start/End) com SSH conector e Tranfer Files, para transferir os vários ficheiros disponibilizados no servidor FTP (um CSV por dia com as leituras do dia anterior, num total de sete ficheiros).

- Na fase Transform (T), foi preparado inicialmente um ciclo para gerar um CSV, consolidando todos os dados disponíveis num único CSV, num processo que lista os ficheiros transferidos e os lê, um por iteração, dentro do loop. No final é criado um CSV consolidado com os dados em bruto para guardar como histórico. A partir desses dados consolidados, damos início ao processo de transformação com o objetivo de transformar os dados iniciais (15 em 15 minutos) em caudais diários.

- Na fase de Load (L), vai gerar um excel com os caudais diários de todos os sensores, e também um outro excel, gerado dentro do loop, agrupando em folhas separadas, os caudais diários de cada sensor. Nesta fase, é ainda implementada uma integração com uma API de dados meteorológicos para obter a precipitação diária nos dias correspondentes, para analisar e relacionar com os dados do caudal diário de cada sensor. No final são gerados gráficos para ajudar a analisar a informação.

Este fluxo de ETL possibilita uma análise consistente, automática e facilmente atualizável, permitindo incluir os ficheiros de entrada atualizados todos os dias.

Processo ETL

A implementação do processo ETL foi efetuada em várias etapas distintas, desde a extração dos dados até à geração dos resultados finais.

Extração (E)

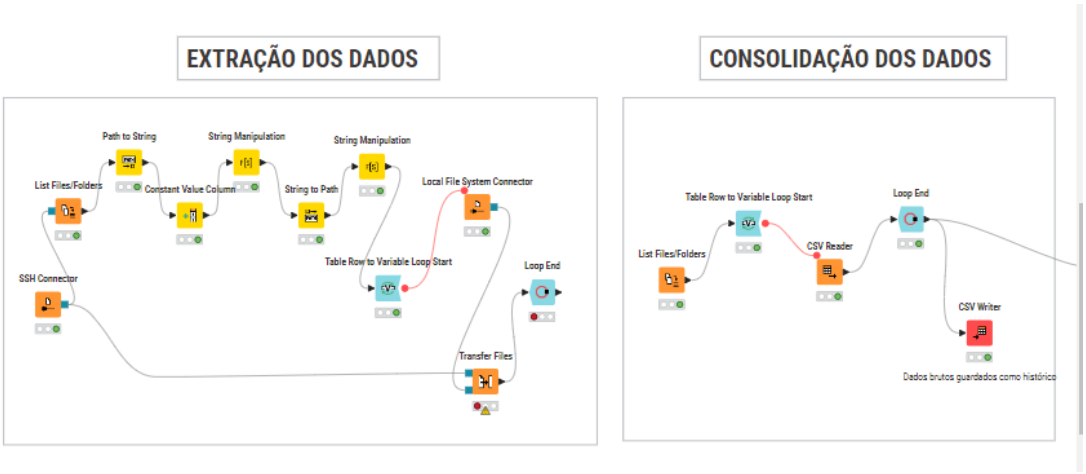

Figura 1 - Processo de Extração e Consolidação dos dados

O processo inicia-se com a extração de dados brutos provenientes dos ficheiros CSV distintos, disponibilizados por FTP:

1: File/Folder List | Flow Variables

Rows: 7 | Columns: 1

#	RowID	Path (csv Path)
1	Row0	/CSV/Lecturas_20251013080748.csv
2	Row1	/CSV/Lecturas_20251014080934.csv
3	Row2	/CSV/Lecturas_20251015081632.csv
4	Row3	/CSV/Lecturas_20251016080901.csv
5	Row4	/CSV/Lecturas_20251017080750.csv
6	Row5	/CSV/Lecturas_20251018080923.csv
7	Row6	/CSV/Lecturas_20251019080827.csv

Figura 2 - Ficheiros CSV disponibilizados por FTP

Estes CSV representam a base do projeto e são a matéria-prima para as transformações posteriores. Cada um destes ficheiros é processado através de um loop de entrada (“Loop Start/End”), que faz a transferência de cada um deles para a pasta de destino em cada iteração. Este processo de transferência falhou por incompatibilidade critica do software, iniciando a fase de consolidação com a listagem direta dos ficheiros no local file system (*C:\TELEMETRIA\CLIENTS*), lidos dentro do loop, um por cada iteração, concatenando toda a informação num único ficheiro CSV (*file1.csv*).

Transformação (T)

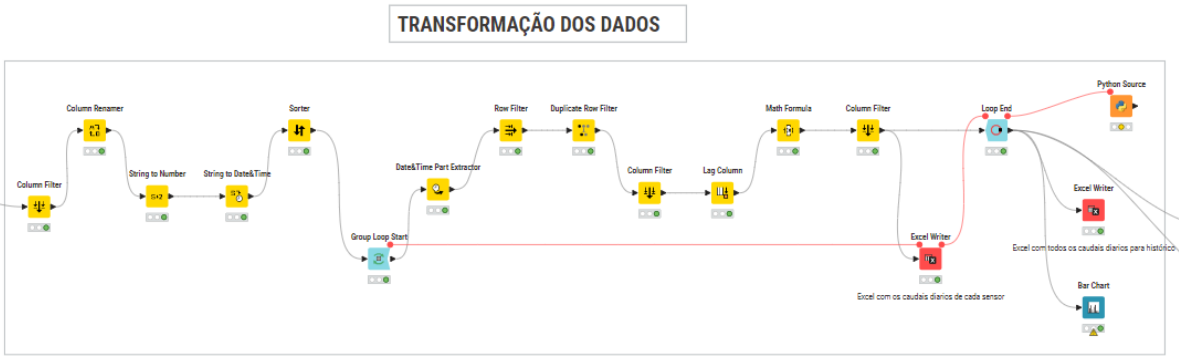

Figura 3 - Processo de Transformação dos dados

Limpeza, Conversão de tipos e Ordenação

Esta fase inicial concentra-se em estruturar e validar os dados brutos, que chegam concatenados de múltiplos ficheiros CSV, preparando-os para os cálculos de séries temporais. Com os dados já consolidados, durante a transformação e mesmo após a sua integração, é necessário fazer uma limpeza e padronização dos campos, de forma a remover informações desnecessárias e uniformizar os nomes das colunas:

- **Column Filter** e **Column Renamer**: Seleção e Padronização: Remove colunas redundantes ou lixo e atribui nomes claros (ID_SENSOR, DATA_HORA, LEITURA) às colunas brutas, facilitando a rastreabilidade do processo.

- **String to Number**: Conversão de Tipo: Converte a coluna LEITURA (valor acumulado) para o tipo numérico (Double/Float).

- **String to Date&Time**: Conversão de Tipo: Converte a coluna DATA_HORA para o formato Date&Time (Local). Esta padronização é essencial para a extração do componente de tempo e para a ordenação correta.

- **Sorter**: Ordenação por Grupo: ordena os dados por ID_SENSOR (Chave de Agrupamento) e DATA_HORA. Esta ordenação rigorosa garante que o cálculo do Lag (registo anterior) seja feito na sequência temporal correta para cada sensor.

Cálculo do Consumo Diário

Esta secção isola os registos de meia-noite (00:00:00) e calcula o consumo do período - caudal diário de cada sensor:

- **Group Loop Start**: Job Control: Inicia um ciclo de execução para cada sensor único (ID_SENSOR), permitindo o processamento e escrita individual por folha.
- **Date&Time Part Extractor**: Extração de Hora: Extrai os componentes de Hour e Minute da coluna DATA_HORA.
- **Row Filter** e **Duplicate Row Filter**: Validação e Filtro Diário: Filtra os dados para isolar apenas os registos de 00:00:00 (Hour = 0, Minute = 0). O Duplicate Row Filter assegura que não há registos repetidos.
- **Column Filter**: Limpeza Intermédia: Remove as colunas temporárias (Hour, Minute) antes do cálculo final.
- **Lag Column**: Cálculo da Leitura Anterior: Dentro do grupo de sensores (Group Loop), cria uma coluna (LEITURA(-1)) com o valor acumulado do dia anterior.
- **Math Formula**: Cálculo final: Executa a subtração ($LEITURA$ - $LEITURA(-1)$), resultando no CONSUMO_DIARIO.
- **Column Filter e Loop End**: Limpeza final: Remove colunas temporárias e o Loop End consolida os resultados do log de execução.

Load (L)

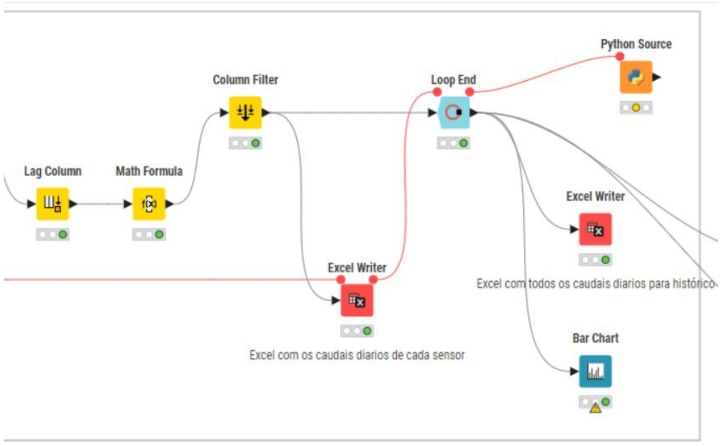

Figura 4 - Processo de Load

Orquestração da escrita e exportação

O Job principal que é orquestrado pelo nó Group Loop Start, e atua como o motor de controlo do fluxo de trabalho automatizado para processar e armazenar o resultado de cada sensor individualmente, que dá origem á exportação dos dados:

- **Group Loop Start**: Job Control: Agrupa os dados por ID_SENSOR e itera sobre cada grupo, permitindo o processamento e escrita individual por folha.
- **Excel Writer**: Load Orquestrado e Divisão por Folha: Posicionado dentro do Job, é configurado para usar a variável de fluxo (ID_SENSOR) para nomear cada folha do Excel. O resultado é exportado para o Excel Consumos_diarios_all.xlsx para o destino no Local File System (C:\TELEMETRIA\CLIENTS\OUTPUT).
- **Python Source**: Desenvolvimento e Integração de Sistemas: Recebe o ficheiro Excel recém-escrito, e o script interno utiliza a biblioteca OpenPyXL para ajustar os cabeçalhos, a largura das colunas e o alinhamento.
- **Loop End**: Fecho do Job: Sinaliza o final da iteração, permitindo que o processo avance para o sensor seguinte, finalizando com o último.
- **Excel Writer (2)**: Exportação de Log: No final do ciclo e após a transformação dos dados estar completa, é gerado um novo Excel com todos os dados dos caudais diários de todos os sensores Consumos_diarios.xlsx que também é exportado para o destino no Local File System (C:\TELEMETRIA\CLIENTS\OUTPUT), guardado como log, para histórico de dados de consumos diários.

Enriquecimento de Dados com Integração de API

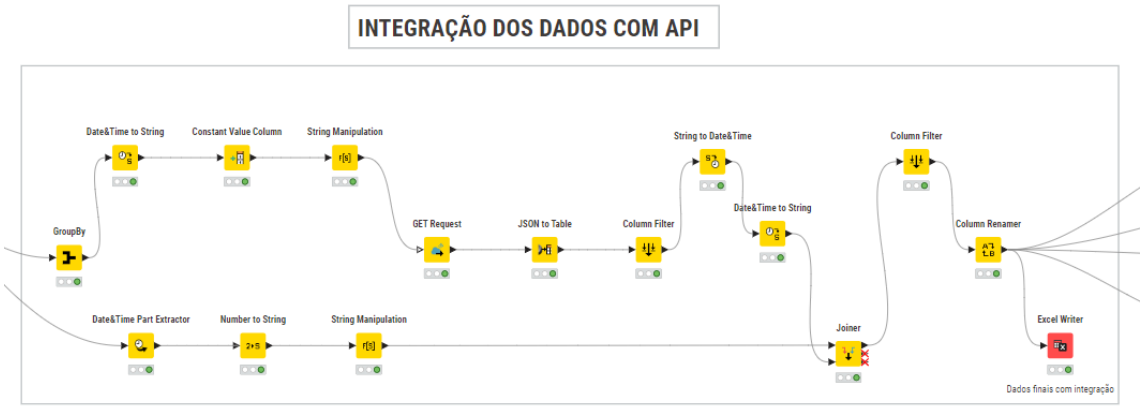

Figura 5 - Integração de dados com API

A integração de API/JSON é um Job paralelo que enriquece os dados, usando o GroupBy e a construção dinâmica de URL. Este processo demonstra uma interligação com um sistema externo e o objetivo deste Job é obter o valor diário de Precipitação de uma API externa (Open-Meteo) para cada data na tabela de consumo e, em seguida, juntar esta informação aos dados de caudal.

- **GroupBy**: Otimização: Recebe a tabela de consumo diário e agrupa-a por DATA_HORA, criando uma lista de datas únicas. Isto evita chamar a API múltiplas vezes para o mesmo dia.

- **Date&Time to String**, **Constant Value Column** e **String Manipulation**: Construção do URL Dinâmico: Esta sequência cria o URL de query de precipitação para cada data única. O String Manipulation é a chave: junta o prefixo estático da API (URL_BASE) com a data, que é convertida para o formato de string YYYY-M-D, conforme exigido pela API.

- **GET Request**: Chamada Externa: Envia a requisição HTTP para a URL dinâmica de cada dia, recebendo o corpo da resposta em formato JSON.

- **JSON to Table**: Transformação Semiestruturada: Recebe a coluna de texto JSON e, através do mapeamento de JSON Path, extrai os valores desejados (como daily.precipitation_sum) e os converte em colunas de uma tabela.

- **Column Filter** e **Column Renamer**: Limpeza Intermédia: Seleciona as colunas essenciais (daily.time.0 e daily.precipitation_sum.0) e as renomeia para clareza (ex: PRECIPITAÇÃO).

- **String to Date&Time** e **Date&Time to String**: Padronização da Chave (Correção de Incompatibilidade): Esta sequência é crucial. Ela converte a data da API (daily.time.0) para o formato de String sem zeros à esquerda (YYYY-M-D) que a sua chave de consumo principal utiliza, resolvendo a incompatibilidade de tipos (Date vs. String) no Joiner.

- **Joiner**: Enriquecimento: Junta a tabela principal de Consumo Diário (proveniente do Math Formula) com a tabela de Metadados Enriquecidos (Precipitação). A junção é feita com base na coluna Data Padronizada (CHAVE_DATA_JOIN no lado esquerdo e a sua coluna correspondente no lado direito) , utilizando sempre o tipo INNER JOIN para garantir que apenas os dados com correspondência em todas as tabelas são incluídos.

- **Joiner**: Enriquecimento: Junta a tabela principal de Consumo Diário (proveniente do Math Formula) com a tabela de Metadados Enriquecidos (Precipitação).

- **Column Filter** e **Column Renamer**: Limpeza Intermédia: Seleciona as colunas essenciais (daily.precipitation_sum.0) e a renomeia para clareza (PRECIPITAÇÃO).

- **Excel Writer**: Output da tabela final enriquecida: Posicionado depois do Job, é o Load final dos dados agrupados com integração. O resultado é exportado para o Excel Resultado final com integração.xlsx para o destino no Local File System (C:\TELEMETRIA\CLIENTS\OUTPUT).

Demonstração Final dos Resultados (Visualização e Análise)

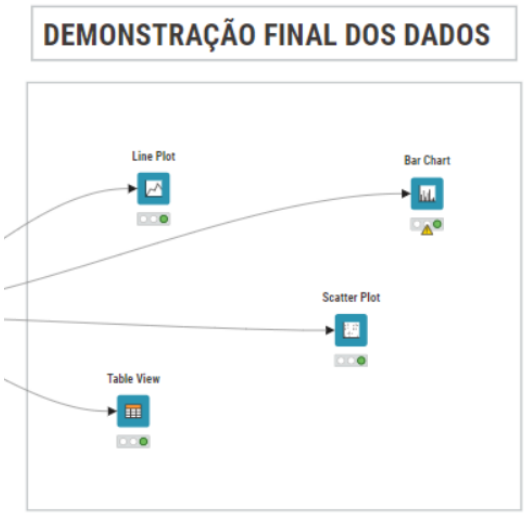

Figura 6 - Demonstração final dos resultados

Esta fase final recebe a tabela enriquecida (com CONSUMO_DIARIO e PRECIPITAÇÃO) que sai do nó Joiner e apresenta os resultados através de gráficos interativos. A união destes views no KNIME constitui o dashboard final do projeto:

- **Line Plot**: Análise de Tendência: Visualizar a evolução do CONSUMO_DIARIO ao longo do tempo (série temporal).
- **Bar Chart**: Comparação de Agregação: Apresentar a precipitação total ou média por mês/ano, ou o consumo total de cada sensor.
- **Scatter Plot**: Análise de Correlação: Investigar a relação entre os caudais diários e a precipitação, analisando a sua implicação direta.
- **Table View**: Evidência Bruta (Controlo de Qualidade): Apresentar a tabela final, permitindo inspecionar visualmente as colunas CONSUMO_DIARIO e PRECIPITAÇÃO lado a lado, confirmando que a junção funcionou e os valores estão preenchidos.

Conclusão e trabalhos futuros

Conclusão

O projeto demonstrou com sucesso a construção de um pipeline completo de Extract, Transform e Load (ETL), focado na Integração de Sistemas de telemetria de caudal. Os objetivos centrais foram integralmente alcançados:

- ✓ **Transformação Complexa**: O pipeline conseguiu replicar e otimizar a lógica de cálculo do consumo diário, superando as limitações iniciais dos dados brutos e as incompatibilidades de tipo (String vs. Date) através de uma cadeia de operadores KNIME (Sorter, Lag Column, Math Formula).

- ✓ **Integração de Sistemas**: Foi demonstrada a capacidade de consumir uma API externa (JSON), enriquecendo os dados de consumo com a precipitação, e resolvida a incompatibilidade de chaves de junção entre os sistemas.

- ✓ **Orquestração**: O uso do Group Loop permitiu implementar uma estrutura de Jobs que automatiza o processamento e o Load dos dados, segmentando a saída final do Excel por sensor (folha de cálculo).

- ✓ **Contorno de Problemas de Software**: O projeto evidenciou a capacidade de desenvolver soluções de contorno para limitações do software (como a restrição do nó String Manipulation e a falta de nós de controlo do Knime), utilizando uma combinação de nós nativos e integração de código Python.

Em suma, o workflow criado é um exemplo robusto de como os processos de ETL podem ser desenhados para limpar, validar e enriquecer dados heterogéneos, preparando-os para análise de negócio.

Trabalhos futuros

Para evoluir este pipeline de ETL de um ambiente de desenvolvimento para um sistema de produção, identificam-se os seguintes trabalhos futuros, focados em automatização, persistência de dados e visualização avançada:

- ✓ **Agendamento e Automação**: Configurar a programação automática do Job principal. O workflow deve ser agendado para ser executado uma vez por dia, às 09:00h, garantindo que os novos dados de telemetria sejam processados e que a transferência dos ficheiros de origem do FTP seja executada automaticamente (Transfer Files e SSH Connector).

- ✓ **Persistência de Dados**: Incluir um output com integração direta em Base de Dados SQL (Ex: MySQL, PostgreSQL). Isto envolve incluir um nó DB Writer e definir um esquema de base de dados relacional (Ex: Normalização para 3FN), garantindo que os dados enriquecidos sejam persistidos de forma estruturada.

- ✓ **Visualização Profissional (Dashboard):** Integrar o output da Base de Dados SQL com a plataforma Grafana. O Grafana permitiria construir dashboards de visualização de resultados altamente profissionais e em tempo real, utilizando a coluna DATA_HORA para gráficos de séries temporais e o CONSUMO_DIARIO para métricas de KPI.

Em síntese, a evolução deste projeto de ETL focar-se-á na automatização total e na escalabilidade. O desenvolvimento futuro visa transicionar o pipeline para um ambiente de produção através da programação automática da transferência e processamento de dados FTP, da persistência estruturada do resultado na Base de Dados SQL, e da integração com o Grafana para a criação de dashboards avançados em tempo real. Estes passos são cruciais para transformar o workflow de desenvolvimento num sistema de inteligência de negócio robusto e operacional.

Bibliografia

https://www.knime.com/etl-software

https://docs.knime.com/

https://restfulapi.net/json-jsonpath/

https://api.open-meteo.com/v1/