

Clima e Consumo HVAC

Integração de Sistemas de Informação

ENRIQUE RODRIGUES Nº28602

Instituto Politécnico do Cávado e do Ave

17 de outubro de 2025

Palavras-chave: Node-RED, Pentaho Kettle, Integração de Dados, IoT, Simulação, HVAC, Modelação Térmica, ETL, Python, pandas, matplotlib

Índice

| | | |
|----------|---|-----------|
| 1 | Enquadramento | 2 |
| 2 | Problema | 3 |
| 3 | Estratégia Utilizada | 4 |
| 3.1 | Aquisição e Simulação (Node-RED) | 4 |
| 3.2 | Modelação Matemática das Variáveis Simuladas | 4 |
| 3.3 | Transformação e Enriquecimento (Pentaho Kettle) | 5 |
| 4 | Transformações e Jobs (Pentaho) | 6 |
| 4.1 | Transformações | 6 |
| 4.2 | Jobs | 7 |
| 5 | Análise e Visualização (Python) | 8 |
| 6 | Vídeo de Demonstração | 9 |
| 7 | Conclusão e Trabalhos Futuros | 10 |

1 Enquadramento

O presente trabalho descreve o desenvolvimento de um pipeline de dados completo para recolha, transformação, armazenamento e análise de variáveis ambientais, nomeadamente: temperatura exterior, temperatura interior simulada e consumo energético de um sistema HVAC.

O sistema é composto por quatro módulos principais:

- **Node-RED:** aquisição de dados e simulação das variáveis ambientais;
- **Pentaho Data Integration (Kettle):** transformação, limpeza, enriquecimento e exportação dos dados;
- **Base de Dados SQLite:** armazenamento estruturado e persistente dos resultados processados;
- **Python (pandas, matplotlib, seaborn):** análise estatística e visualização gráfica dos resultados.

2 Problema

A gestão eficiente da climatização em edifícios residenciais e comerciais é um desafio crescente, dada a variabilidade das condições exteriores e o impacto direto no consumo energético. O controlo manual do HVAC ou a falta de monitorização contínua leva a desperdício de energia e a condições de conforto inconsistentes.

O objetivo deste trabalho é demonstrar um pipeline de dados capaz de:

- Recolher temperaturas exteriores via API pública, permitindo monitorização contínua;
- Simular a evolução da temperatura interior e o consumo do HVAC, com base em modelos térmicos iterativos;
- Processar, validar e enriquecer os dados no Pentaho, incluindo cálculos de eficiência e diferença térmica;
- Exportar os resultados para vários formatos — **CSV, XML e Base de Dados SQLite** — de modo a permitir integração e análise em diferentes plataformas;
- Gerar gráficos e análises estatísticas em Python que permitam avaliar padrões de consumo e identificar oportunidades de otimização energética.

Desta forma, o pipeline não só replica o comportamento térmico de um edifício como também fornece dados estruturados, persistentes e prontos para análise, permitindo uma avaliação clara da eficiência do sistema HVAC e do impacto das condições ambientais.

3 Estratégia Utilizada

3.1 Aquisição e Simulação (Node-RED)

Os dados de temperatura exterior são obtidos via API REST. A simulação da temperatura interior e do consumo do HVAC é realizada em JavaScript. Cada sala tem parâmetros térmicos próprios (α , β).

3.2 Modelação Matemática das Variáveis Simuladas

Esta secção descreve os modelos utilizados para simular a evolução térmica das divisões e o consumo energético do sistema HVAC. Os modelos foram implementados em JavaScript, correndo em ciclo a partir da temperatura anterior e das condições exteriores obtidas por API. Os parâmetros α e β variam por divisão e representam, respetivamente, o grau de isolamento e a capacidade de resposta do sistema de climatização.

Evolução da Temperatura Interior

A temperatura interior de cada divisão evolui de forma gradual, aproximando-se da temperatura exterior devido às trocas térmicas do edifício e da temperatura de conforto devido à atuação do HVAC. A equação seguinte modela esta dinâmica:

$$T_{\text{interior}}(t) = T_{\text{anterior}} + \alpha \cdot (T_{\text{exterior}} - T_{\text{anterior}}) + \beta \cdot (T_{\text{conforto}} - T_{\text{anterior}}) + \epsilon$$

Onde:

- $T_{\text{interior}}(t)$: temperatura interior no instante atual;
- T_{anterior} : temperatura interior no instante anterior;
- T_{exterior} : temperatura exterior (obtida por API);
- T_{conforto} : temperatura de conforto (21°C);
- α : coeficiente de isolamento térmico do edifício;
- β : coeficiente de correção aplicado pelo HVAC;
- ϵ : ruído aleatório que introduz variação realista.

Este modelo é iterativo - a cada ciclo de simulação, a temperatura interior é atualizada em função dos valores anteriores. A componente α controla a velocidade de dissipação térmica (edifícios mais isolados têm α menor), enquanto β representa a eficiência de regulação térmica do HVAC.

Consumo Energético do HVAC

O consumo energético é proporcional ao esforço necessário para manter o conforto térmico. Quanto maior a diferença entre a temperatura exterior e a de conforto, maior é a energia necessária para compensar as perdas ou ganhos térmicos. A equação seguinte define este comportamento:

$$P_{\text{HVAC}} = \max \left(20, 50 + 5 \cdot |T_{\text{conforto}} - T_{\text{exterior}}| + 20 \cdot |T_{\text{interior}} - T_{\text{conforto}}| + \eta \right)$$

Onde:

- P_{HVAC} : consumo energético instantâneo (W);
- $|T_{\text{conforto}} - T_{\text{exterior}}|$: esforço devido às condições exteriores;
- $|T_{\text{interior}} - T_{\text{conforto}}|$: esforço interno para manter o conforto;
- η : ruído aleatório para simular variações de carga;
- $\max(20, \cdot)$: garante um consumo mínimo de 20 W (modo de *standby*).

Este modelo cria uma relação linear simplificada entre o desvio térmico e o consumo energético.

3.3 Transformação e Enriquecimento (Pentaho Kettle)

Os dados simulados no Node-RED são posteriormente processados no Pentaho. O objetivo é normalizar os registos, limpar os nomes e derivar métricas adicionais de desempenho térmico.

- **CSV Input:** leitura dos dados gerados pelo Node-RED;
- **Regex Replace:** limpeza e padronização de nomes de dispositivos e salas;
- **Calculator:** criação de variáveis derivadas para análise de eficiência:

$$\text{temp_diff} = |T_{\text{exterior}} - 21| \quad \text{e} \quad \text{efficiency_index} = \frac{\text{avg_power}}{(\text{temp_diff} + 1)}$$

Estas métricas permitem avaliar o impacto da diferença térmica no consumo e comparar o desempenho entre divisões.

- **CSV/XML Output:** exportação dos dados tratados e enriquecidos.

O resultado é um conjunto de ficheiros estruturados prontos para análise estatística e visualização em Python.

4 Transformações e Jobs (Pentaho)

4.1 Transformações

A transformação principal, designada “**etl_climate_analysis**”, foi desenvolvida no Pentaho Data Integration (Kettle) e tem como objetivo processar os ficheiros CSV gerados pelo Node-RED, realizar operações de limpeza e enriquecimento, e exportar os resultados para múltiplos destinos.

A transformação é composta pelas seguintes etapas principais:

- **CSV File Input:** Lê os dados provenientes do Node-RED. Cada linha contém o timestamp, o identificador do dispositivo, o nome da divisão, o tipo de sensor, o valor medido e a temperatura exterior correspondente. Os campos são importados com os respetivos tipos (string, number, date).
- **Validate Timestamp (Regex):** Validação da coluna de data/hora através de uma expressão regular, garantindo o formato correto antes do processamento. Linhas com timestamps inválidos são rejeitadas.
- **Filter Rows (Sensor Type = Power):** Filtra apenas os registos relativos ao consumo energético (`sensor_type = "power"`), eliminando medições de temperatura interna. Este passo reduz o volume de dados e foca a análise no comportamento energético do sistema HVAC.
- **Group By (Aggregate Power):** Os dados são agregados por **timestamp** e **divisão**, calculando:
 - `avg_power` — consumo médio de energia;
 - `outdoor_temp` — média da temperatura exterior;
- **Formula Step (Calculated Fields):** Através da etapa *Formula*, são geradas novas colunas derivadas:
 - `efficiency` — eficiência térmica estimada com base no rácio entre potência média e diferença térmica;
 - `temp_diff` — diferença absoluta entre temperatura interior simulada e exterior.

Estes cálculos complementam a análise energética do sistema HVAC.

- **Table Output (SQLite):** Os dados enriquecidos são inseridos numa base de dados SQLite local (`hvac_data.db`).

- **Text File Output (CSV Export):** Exporta o dataset final para formato CSV, permitindo integração com outras ferramentas analíticas.
- **XML Output:** Gera um ficheiro XML com a mesma informação, facilitando a interoperabilidade e validação estrutural dos dados.

O fluxo completo garante a consistência entre diferentes formatos de exportação e cria um dataset limpo e consolidado para análise posterior.

4.2 Jobs

O job, denominado “**etl_hvac_power**”, orquestra a execução das transformações e gere o fluxo global de execução.

As etapas principais são:

- **Log Start:** Gera uma entrada de log com a data/hora de início, permitindo o acompanhamento das execuções no histórico de logs.
- **Run Transformation (climate_transform):** Chama a transformação principal, realizando todo o processo de ETL descrito anteriormente. Caso ocorram erros, o job é interrompido e o resultado é registado como falha.
- **Log End:** Regista a conclusão da execução e o estado final (sucesso ou erro), encerrando o processo de forma controlada.
- **Email:** Manda um email para confirmar o fim do job com os logs.

Este job garante a execução automatizada e sequencial das tarefas.

5 Análise e Visualização (Python)

Após a execução do pipeline ETL, os ficheiros finais são processados em Python para análise estatística e visualização gráfica dos dados. As bibliotecas utilizadas incluem:

- pandas ($v \geq 2.0$) para manipulação, limpeza, agregação e fusão de datasets;
- matplotlib ($v \geq 3.7$) para a criação de gráficos básicos de linhas, barras e dispersão;
- seaborn ($v \geq 0.13$) para visualizações estatísticas avançadas, incluindo box-plots e regressões lineares.

Foram produzidos os seguintes gráficos a partir do dataset processado:

- **Scatter Plot — Power vs Outdoor Temperature:** visualiza a relação entre a temperatura exterior e o consumo médio de energia (avg_power). (Figura: power_vs_outdoor_temp.png)
- **Scatter Plot — Power vs Temperature Difference:** mostra a relação entre a diferença de temperatura (temp_diff) e o consumo médio de energia. (Figura: power_vs_temp_diff.png)
- **3D Scatter Plot — Outdoor Temp vs Temp Diff vs Avg Power:** fornece uma visão tridimensional do consumo energético em função da temperatura exterior e da diferença de temperatura, codificando a eficiência pelo esquema de cores. (Figura: 3d_power_temp.png)
- **Histogram of Efficiency:** distribuição da eficiência do HVAC ao longo de todos os registos, permitindo identificar padrões e outliers. (Figura: efficiency_hist.png)
- **Boxplot of Avg Power per Room:** análise comparativa do consumo médio de energia por sala, destacando variações entre divisões. (Figura: avg_power_boxplot.png)

Estas visualizações permitem identificar padrões de consumo, períodos de maior esforço do HVAC, diferenças entre salas e a relação entre condições exteriores e desempenho energético. A utilização conjunta de scatter plots, 3D plots, histogramas e boxplots fornece uma análise completa e facilmente interpretável do comportamento do sistema HVAC, complementando os cálculos realizados no Pentaho.

6 Vídeo de Demonstração

Para complementar o relatório, foi produzido um vídeo demonstrativo do pipeline completo, incluindo:

- Simulação de dados com Node-RED;
- Processamento e transformação no Pentaho (Kettle);
- Exportação de dados para CSV, XML e SQLite;
- Visualização e análise em Python.

O vídeo pode ser acessado diretamente através do seguinte QR Code:



Link direto: <https://drive.google.com/file/d/1q7uXCaNz5Eu7ypg1x0YI0l0xh1eN9QmP/view>

7 Conclusão e Trabalhos Futuros

O presente trabalho permitiu desenvolver e implementar um pipeline completo de ETL para monitorização e análise de sistemas HVAC, desde a recolha de dados até à visualização de resultados. Foram alcançados os seguintes objetivos:

- Integração de dados simulados e reais via Node-RED, combinando informação de sensores internos e dados meteorológicos externos;
- Processamento e limpeza de dados no Pentaho, incluindo validação de timestamps, filtragem, agregações e cálculo de métricas derivadas (*efficiency* e *temp_diff*);
- Exportação de dados para vários formatos (CSV, XML e SQLite) garantindo persistência e facilidade de análise;
- Visualização e análise estatística em Python, permitindo identificar padrões de consumo energético, comportamento térmico das divisões e eficiência do sistema HVAC.

O pipeline demonstrou ser reproduzível, modular e escalável, fornecendo *insights* relevantes para monitorização e otimização energética de edifícios.

Trabalhos Futuros:

- Implementação de logging mais detalhado e métricas de desempenho do ETL, permitindo monitorização contínua do pipeline;
- Expansão da simulação para incluir outros parâmetros ambientais, como humidade, ocupação e consumo real de energia;

Estas melhorias permitiriam transformar o pipeline atual num sistema inteligente de monitorização e otimização de climatização, apto a utilização em contextos reais de edifícios residenciais ou comerciais.

Referências

- [1] Matplotlib Documentation. <https://matplotlib.org/stable/contents.html>. Accessed: 2025-10-16.
- [2] Node-RED Documentation. <https://nodered.org/docs/>. Accessed: 2025-10-16.
- [3] Open-Meteo. Open-Meteo Historical Weather API. <https://open-meteo.com/en/docs/historical-weather-api>. Accessed: 2025-10-16. 2025.
- [4] Python pandas Documentation. <https://pandas.pydata.org/docs/>. Accessed: 2025-10-16.
- [5] Seaborn Documentation. <https://seaborn.pydata.org/>. Accessed: 2025-10-16.
- [6] SQLite Documentation. <https://www.sqlite.org/docs.html>. Accessed: 2025-10-16.