_02_Clean_Transform_Final.ktr

Engenharia de Sistemas Informáticos

UC: Integração de Sistemas de Informação 2025-2026

RELATÓRIO DO TRABALHO PRÁTICO 01

Nome	Nº de Aluno
Gonçalo Santos	27985

19 de outubro de 2025

Conteúdo

Lista de Figuras

1 Enquadramento

Este relatório apresenta a análise e implementação de um processo de **Extract, Transform, Load (ETL)** desenvolvido no âmbito da unidade curricular de *Integração de Sistemas de Informação (ISI)* da Licenciatura em Engenharia de Sistemas Informáticos do Instituto Politécnico do Cávado e do Ave (IPCA).

O projeto foca-se na integração de dados oceanográficos e de informações sobre praias, com o objetivo de criar uma base de dados analítica denominada `marine_esposende_dw`, que suporte a monitorização e análise de condições marinhas na região de Esposende.

Este trabalho visa **consolidar conceitos associados à Integração de Sistemas de Informação usando dados** e **explorar ferramentas de suporte a processos de ETL**, conforme os objetivos definidos no enunciado do trabalho prático (Instituto Politécnico do Cávado e do Ave, 2025). Em particular, este projeto demonstra a aplicação de diversos critérios de mais-valia, tais como a **utilização de Expressões Regulares**, o **tratamento de múltiplos formatos de dados** (XML, JSON, CSV), o **desenvolvimento de jobs completos**, a aplicação de **joins e lookups**, uma vasta gama de **operações sobre valores**, **geração de logs**, **acesso a APIs remotas**, **operações essenciais sobre bases de dados** e a **visualização dos resultados em dashboards**.

A crescente necessidade de **tomada de decisão baseada em dados** em diversos setores — incluindo a gestão ambiental e a segurança costeira — sublinha a importância de sistemas robustos de integração de dados.

Deste modo, este trabalho prático visa explorar as capacidades de ferramentas ETL, nomeadamente o *Pentaho Data Integration (Kettle)*, para construir um fluxo de dados que extrai informações de fontes heterogéneas, aplica transformações de limpeza e enriquecimento, e carrega os dados para um *data warehouse*.

2 Problema

O problema central abordado neste trabalho é a **integração e disponibilização de dados oceanográficos e geográficos de praias** para análise, um cenário que reflete a crescente complexidade dos sistemas de informação e a necessidade de *insights* acionáveis (Instituto Politécnico do Cávado e do Ave, 2025).

Este desafio multifacetado abrange várias dimensões críticas, alinhadas com os critérios de mais-valia destacados no enunciado (Instituto Politécnico do Cávado e do Ave, 2025):

- **Extração de Múltiplas Fontes:** O projeto exigiu a extração de dados de fontes heterogéneas, nomeadamente uma API de dados marinhos dinâmicos (`marine-api.open-meteo.com`) e um ficheiro CSV estático (`Praias.csv`) com informações de praias, demonstrando a capacidade de lidar com a **diversidade de formatos de representação de dados** e **explorar o acesso a APIs remotas** (Instituto Politécnico do Cávado e do Ave, 2025).

- **Qualidade dos Dados:** A garantia da qualidade dos dados foi um pilar fundamental. Tal foi alcançado através de validação rigorosa, normalização e limpeza, com a **utilização proeminente de Expressões Regulares (ER)** para padrões de validação e a implementação de processos de **normalização de dados** para assegurar a consistência (Instituto Politécnico do Cávado e do Ave, 2025).

- **Enriquecimento de Dados:** Os dados foram enriquecidos significativamente através de **operações sobre valores** complexas, incluindo o cálculo de métricas derivadas (por exemplo, energia das ondas e velocidade da corrente), a classificação de condições ambientais (estado das ondas, intensidade das correntes, classes de temperatura da água) e a deteção de anomalias, demonstrando a **diversidade de operadores de transformação** (Instituto Politécnico do Cávado e do Ave, 2025).

- **Prevenção de Duplicação:** A integridade dos dados foi assegurada pela prevenção ativa de duplicação de registos, garantindo que apenas dados novos e únicos fossem carregados para o *data warehouse*. Para tal, foram aplicadas técnicas de **fusão de dados (merging)** e **joins**, que são critérios de mais-valia essenciais (Instituto Politécnico do Cávado e do Ave, 2025).

- **Logging e Alertas:** Foram desenvolvidos mecanismos robustos de **geração de logs** para monitorizar a execução do processo ETL e **acesso a serviços remotos como e-mail** para notificar proativamente sobre anomalias detetadas, o que constitui um aspeto crucial na gestão de *workflows* de dados (Instituto Politécnico do Cávado e do Ave, 2025).

- **Estruturação para Análise:** Os dados foram cuidadosamente estruturados num esquema de *data warehouse* otimizado para consultas analíticas (`marine_esposende_dw`), evidenciando **operações essenciais sobre bases de dados** e a preparação para a **visualização dos resultados conseguidos utilizando dashboards** (Instituto Politécnico do Cávado e do Ave, 2025).

Este problema reflete desafios comuns em projetos de *Business Intelligence* e *Big Data*, onde a integração de dados de diversas origens e a garantia da sua qualidade são cruciais para a obtenção de *insights* fiáveis.

3 Estratégia Utilizada

A estratégia de implementação adotada para resolver o problema centraliza-se na utilização do **Pentaho Data Integration (Kettle)** como ferramenta primária para o desenvolvimento dos processos ETL. A escolha do Pentaho Kettle justifica-se pela sua robustez, flexibilidade e capacidade de lidar com diversas fontes e destinos de dados, além de permitir a implementação de lógicas de transformação complexas sem a necessidade de codificação extensiva, o que se alinha perfeitamente com o objetivo de **explorar ferramentas de suporte a processos de ETL**. O projeto foi estruturado em dois fluxos principais de dados, cuidadosamente orquestrados por um *Job* central, o que demonstra a **orquestração de processos de transformação desenvolvidos** e a capacidade de criar um **projeto completo envolvendo acesso a serviços remotos** (como o envio de e-mails para alertas):

1. **Carregamento e Validação de Dados de Praias (`TR_Carregar_Praias.ktr`):** Esta transformação é um exemplo claro da gestão de **importação/exportação de dados de/-para CSV**, da **utilização de Expressões Regulares (ER)** para validação de nomes de praias, da **normalização de dados** (minúsculas, capitalização) e de **operações essenciais sobre bases de dados** para o carregamento na tabela `marine.fact_beachs`.

2. **Extração, Transformação e Carregamento de Dados Oceanográficos:** Este fluxo demonstra a **exploração do acesso a APIs remotas** (`Trans_01_Extract_Marine_API.ktr`) para a obtenção de dados em formato JSON, a **importação/exportação de dados de/-para XML** (para *staging*), e a aplicação de uma vasta gama de **operações sobre valores** (cálculos, classificações, deteção de anomalias) e **diversidade de operadores de transformação** (`Trans_02_Clean_Transform_Final.ktr`), culminando no carregamento para a tabela `marine.fact_marine_data`.

O *Job* (`Job_Marine_ETL.kjb`) coordena a execução destas transformações, garantindo a **orquestração de processos de transformação desenvolvidos** e a gestão de dependências. A base de dados **PostgreSQL** (`marine_esposende_dw`) é utilizada como *data warehouse*, servindo como destino final para os dados processados e suportando **operações essenciais sobre bases de dados**.

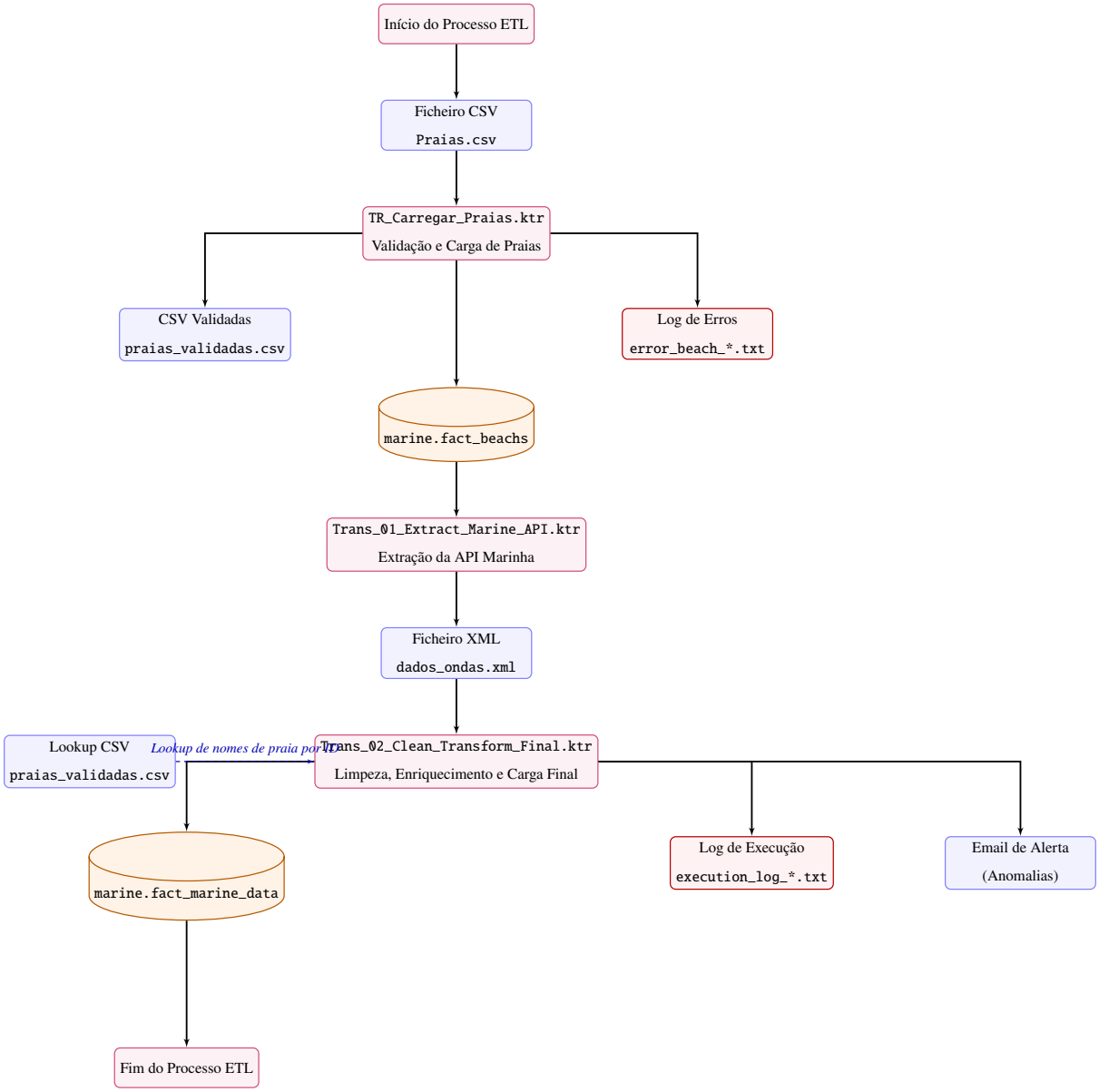

Figura 1: Fluxo geral do processo ETL, incluindo o *lookup* do ficheiro `praias_validadas.csv` na fase de limpeza e enriquecimento.

4 Transformações

Nesta secção são detalhadas as três principais transformações desenvolvidas no âmbito do processo ETL implementado em *Pentaho Data Integration (Kettle)*. Cada transformação foi concebida de modo modular, permitindo a manutenção independente e uma orquestração eficiente no *Job* central. As transformações respeitam as boas práticas de engenharia de dados, tais como a normalização de entradas, a validação sintática, a rastreabilidade de logs e a prevenção de duplicação de registos. (Instituto Politécnico do Cávado e do Ave, 2025)

—

4.1 `TR_Carregar_Praias.ktr` – Carregamento e Validação de Dados de Praias

Esta transformação é responsável por processar os dados iniciais das praias a partir de um ficheiro CSV e carregá-los na tabela `marine.fact_beachs`. O seu principal objetivo é garantir que apenas dados válidos e consistentes sejam persistidos na base de dados, aplicando regras de normalização, validação e deduplicação.

Passos Principais:

1. **Leitura do Ficheiro de Origem:** A transformação inicia com o passo *CSV Input*, que lê o ficheiro `Praias.csv`. Este passo demonstra a capacidade de **lidar com importação/exportação de dados de/para CSV**, conforme os critérios de mais-valia indicados no enunciado.

2. **Validação de Dados com Expressões Regulares:** O passo *Regex Evaluation* aplica a expressão regular `^[A-Za-z- \s]{3,100}$` sobre o campo `beach_name`, assegurando que o nome contém apenas caracteres válidos e um comprimento aceitável. Este procedimento reforça o critério de **validação e normalização de dados**.

3. **Normalização de Atributos Textuais:** Os passos *String Operations* executam normalizações sobre os campos `beach_name` e `orientation`, incluindo conversão para minúsculas e capitalização, eliminando discrepâncias e redundâncias textuais.

4. **Filtragem Condicional:** O passo *Filter Rows* separa registos válidos e inválidos com base no campo `nome_valido`. Os registos válidos seguem para o fluxo principal, enquanto os inválidos são redirecionados para geração de logs.

5. **Registo de Erros e Auditoria:** Os registos inválidos são exportados para o ficheiro `error_beach_*.txt` através do passo *Text File Output*, garantindo a rastreabilidade e o cumprimento do critério de **Geração de Logs**.

6. **Deduplicação e Integração com a Base de Dados:** O passo *Table Input* lê a tabela `marine.fact_beachs` existente, e o *Merge Join* compara os dados atuais com os novos, utilizando como chaves os campos `beach_name`, `lat` e `lon`. Apenas os registos novos e não duplicados são carregados no destino.

7. **Persistência Final:** O passo *Table Output* insere os registos novos na base de dados `PostgreSQL`, concretizando as **operações essenciais sobre bases de dados**.

Além de assegurar a qualidade dos dados de origem, esta transformação produz dois artefactos de apoio fundamentais:

- O ficheiro `praias_validadas.csv`, que serve de base para as transformações subsequentes.
- O ficheiro de log `error_beach_*.txt`, utilizado para auditoria e deteção de erros recorrentes.

Esta abordagem modular reflete as boas práticas de engenharia de sistemas de informação, reforçando a importância de normalizar dados logo nas fases iniciais do ciclo ETL.

4.2 `Trans_01_Extract_Marine_API.ktr` – Extração de Dados da API Marinha

Esta transformação é responsável por extrair dados oceanográficos em tempo real a partir de uma *API RESTful* externa, integrando múltiplas fontes de informação de forma automatizada. O seu principal objetivo é recolher dados ambientais e meteorológicos relevantes para cada

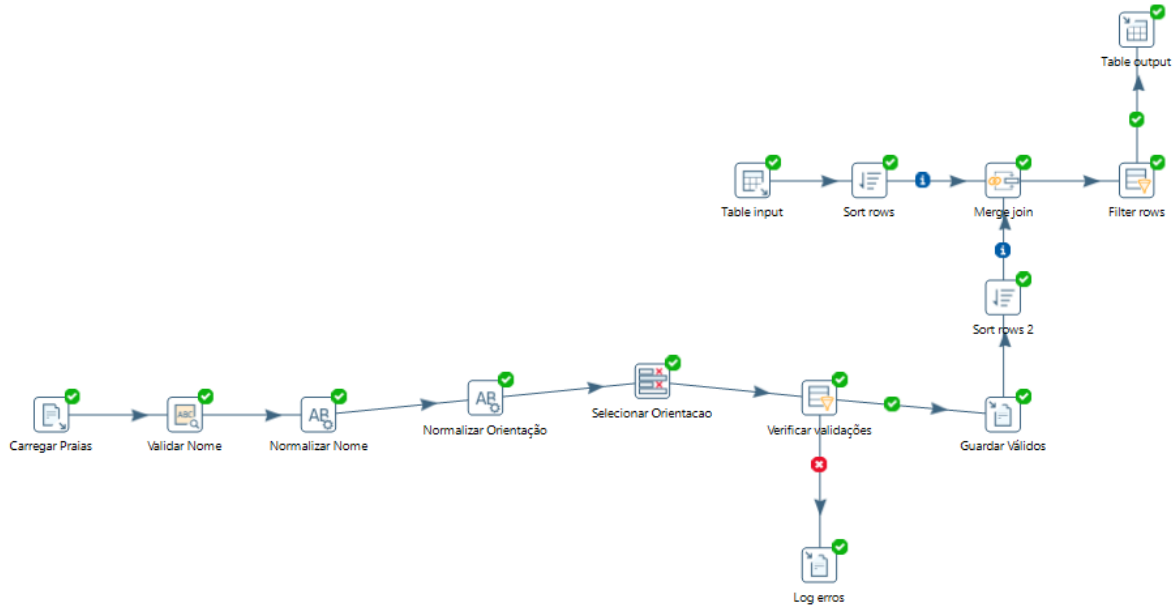

Figura 2: TR_Carregar_Praias.ktr

praia previamente validada, estabelecendo a base para as fases subsequentes de limpeza e enriquecimento.

Passos Principais:

1. **Leitura do Ficheiro de Praias Validadas:** O passo *CSV Input* lê o ficheiro `praias_validadas.csv`, que contém a lista de praias consideradas válidas após a primeira transformação. Este passo demonstra a reutilização de dados previamente processados, garantindo **consistência e integridade de pipeline**.

2. **Construção Dinâmica da URL da API:** A cada iteração, o passo *Modified JavaScript Value* concatena dinamicamente a *string* base da API com os parâmetros de latitude e longitude de cada praia. Este procedimento gera uma URL no formato:

   ```
   https://marine-api.open-meteo.com/v1/marine?latitude=X&longitude=Y
   &hourly=wave_height,wave_direction,water_temperature,current_speed
   &timezone=Europe/Lisbon
   ```

   Esta etapa exemplifica a utilização de **parâmetros dinâmicos** e a adaptação do fluxo ETL a dados geográficos contextuais.

3. **Extração dos Dados via HTTP:** O passo *HTTP Client* executa chamadas GET para cada URL gerada, estabelecendo comunicação segura com o servidor remoto. O manuseamento de *timeouts*, códigos de erro HTTP e reintentos automáticos assegura a **robustez na extração de dados remotos**.

4. **Interpretação e Conversão de JSON:** A resposta recebida em formato JSON é interpretada pelo passo *JSON Input*, que realiza o mapeamento dos campos relevantes (por exemplo, altura de onda, direção das ondas, temperatura da água, velocidade da corrente e pressão atmosférica). Este passo reforça o domínio sobre a **importação/exportação de dados de/para JSON** e a manipulação de dados estruturados.

5. **Filtragem e Pré-Processamento:** O passo *Filter Rows* remove registos incompletos ou com valores nulos, garantindo que apenas dados válidos prosseguem para a área de *staging*. Esta operação assegura que o conjunto de dados resultante cumpre os princípios de qualidade de dados (completude e validade).

6. **Armazenamento Temporário (Área de Staging):** Por fim, o passo *XML Output* armazena o resultado no ficheiro `dados_ondas.xml`, mantendo a estrutura temporal e geográfica dos registos. O formato XML é intencionalmente escolhido por permitir interoperabilidade com transformações posteriores e por suportar estrutura hierárquica complexa.

A transformação `Trans_01_Extract_Marine_API.ktr` demonstra de forma clara a capacidade de **explorar o acesso a serviços remotos via API**, aliando extração dinâmica de dados georreferenciados, integração entre formatos heterogéneos (CSV, JSON, XML) e preparação para **processos de enriquecimento e carga analítica**. A adoção de boas práticas de logging, gestão de falhas e uso de variáveis parametrizadas reflete um design maduro e escalável do ponto de vista de engenharia de dados.

4.3 `Trans_02_Clean_Transform_Final.ktr` – Limpeza, Transformação e Carregamento Final de Dados Marinhos

Esta é a transformação mais complexa e crítica de todo o processo ETL, responsável pela aplicação de regras de negócio, cálculo de métricas avançadas, deteção de anomalias e carregamento

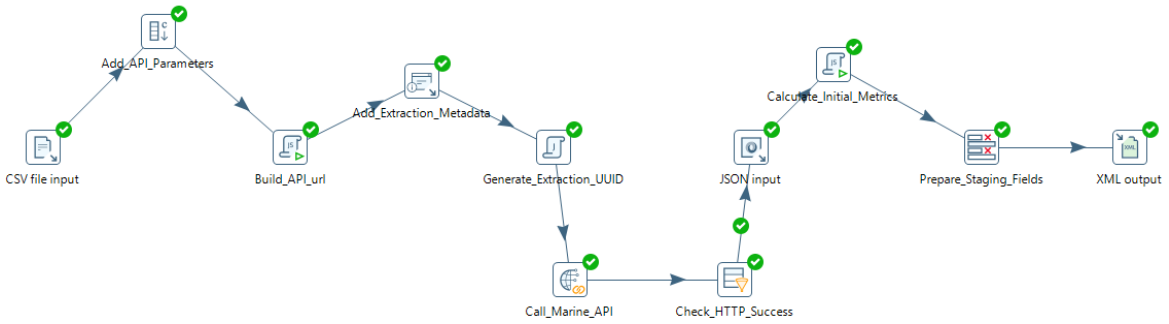

Figura 3: Trans_01_Extract_Marine_API.ktr

final dos dados na base de dados analítica `marine.fact_marine_data`. Esta etapa demonstra a **diversidade de operadores de transformação** disponíveis no Pentaho e a capacidade de realizar **operações sobre valores**, **lookups** e **joins**, consolidando a robustez do pipeline.

Passos Principais:

1. **Registo do Tempo de Execução:** O processo inicia com o passo *System Info*, que cria o campo `processing_start`, registando a data e hora do início do processamento. Este valor é utilizado posteriormente para auditoria e logging, garantindo rastreabilidade temporal das execuções.

2. **Leitura da Área de Staging:** O passo *XML Input* lê o ficheiro `dados_ondas.xml`, gerado na transformação anterior, demonstrando a capacidade de **importar/exportar dados de/para XML**. As linhas com o campo `timestamp_utc` nulo são eliminadas pelo passo *Filter Rows*, assegurando completude dos dados.

3. **Conversão de Tipos e Preparação de Campos:** O passo *Modified JavaScript Value* converte o campo `timestamp_utc` de texto para um objeto de data (`Date`), armazenado no novo campo `timestamp`. Esta conversão garante compatibilidade com os formatos de tempo utilizados no `PostgreSQL`.

4. **Cálculo de Métricas Derivadas:** O passo *JavaScript Value* (*Calculate_Derived_Fields*) calcula indicadores físicos relevantes, nomeadamente:

   - `wave_energy_kj_m2`: Energia das ondas ($E = 0.5 \times \rho \times g \times H^2$), com $\rho = 1025\,kg/m^3$ e $g = 9.81\,m/s^2$;

- `ocean_current_kmh`: Conversão da velocidade da corrente de m/s para km/h;
- `sea_temp_fahrenheit`: Conversão da temperatura da água para Fahrenheit;
- `sea_level_anomaly_m`: Anomalia do nível do mar (referenciado a 0 m).

Estes cálculos demonstram a aplicação de **operações sobre valores** e enriquecimento de dados com base em fórmulas físicas reais.

5. **Classificação de Condições Marinhas:** O passo *JavaScript Value* (*Classify_Conditions*) categoriza as condições marinhas em classes semânticas:

   - `wave_condition`: *Calm*, *Smooth*, *Moderate*, *Rough*, *Very Rough*, *High*;
   - `current_intensity`: *Weak*, *Moderate*, *Strong*, *Very Strong*;
   - `water_temp_class`: *Cold*, *Cool*, *Mild*, *Warm*, *Hot*.

   Esta abordagem permite agregar e simplificar variáveis contínuas, favorecendo a análise exploratória e a futura visualização em *dashboards*.

6. **Cálculo de Indicadores de Qualidade:** O passo *JavaScript Value* (*Calculate_Quality_Metrics*) avalia a integridade e fiabilidade dos dados, calculando:

   - `data_completeness_pct`: Percentagem de campos não nulos;
   - `data_quality_score`: Índice de qualidade global (0–100), ponderando completude e consistência;
   - `is_validated`: Indicador booleano de validação (`true` se `score` ≥ 70).

   Este cálculo materializa a preocupação com a **qualidade dos dados** e garante rastreabilidade analítica.

7. **Deteção de Anomalias:** O passo *JavaScript Value* (*Detect_Anomalies*) identifica padrões anómalos com base em regras de negócio:

   - `wave_height` > 8 m;
   - `water_temperature` < 5°C ou > 30°C;
   - `current_speed` > 4 km/h;

- `sea_level_anomaly` > 2 m ou < -2 m;
- `data_quality_score` < 50.

Registos que satisfaçam estas condições são marcados como anómalos (`is_anomaly = true`), evidenciando a **capacidade de análise e deteção de eventos críticos**.

8. **Lookup de Nomes de Praia:** O passo *Stream Lookup* associa os dados oceanográficos à tabela de referência `praias_validadas.csv`, utilizando o `beach_id` como chave. Este enriquecimento adiciona o campo `beach_name` e representa a aplicação de um **lookup stream** para reforço sem redundância de dados.

9. **Eliminação de Duplicações e Integração com a Base de Dados:** Os dados novos são comparados com os existentes na base de dados através dos passos *Table Input* e *Merge Join*, utilizando as chaves `latitude`, `longitude` e `timestamp_utc`. Apenas os registos não existentes são filtrados pelo passo *Filter Rows* (*Filter_Only_New*) e inseridos no destino, assegurando **integridade e consistência**.

10. **Carga Final de Dados:** O passo *Table Output* (*Load_to_PostgreSQL*) grava os novos registos na tabela `marine.fact_marine_data` da base de dados analítica `marine_esposende_dw`. Esta operação concretiza o encerramento do ciclo ETL e evidencia **operações essenciais sobre bases de dados**.

11. **Geração de Alertas e Logs:** Os registos marcados como anómalos são tratados nos passos *Filter Rows* e *JavaScript Value* (*Alerta*), onde é construído o corpo do e-mail de notificação. As informações são exportadas para um ficheiro `alert/alert_*.txt` e enviadas por e-mail através do passo *Mail*, exemplificando o **acesso a serviços remotos (SMTP)**. O passo final *Text File Output* (*Write to Log*) regista o histórico de execução no ficheiro `execution_log_*.txt`, documentando duplicações e falhas de carga.

A transformação `Trans_02_Clean_Transform_Final.ktr` representa o culminar do processo de integração, unindo extração, validação, cálculo e monitorização num fluxo único e automatizado. O desenho desta transformação reflete a adoção de práticas de **Data Quality Management**, **Data Enrichment** e **Exception Handling**, reforçando a conformidade com os

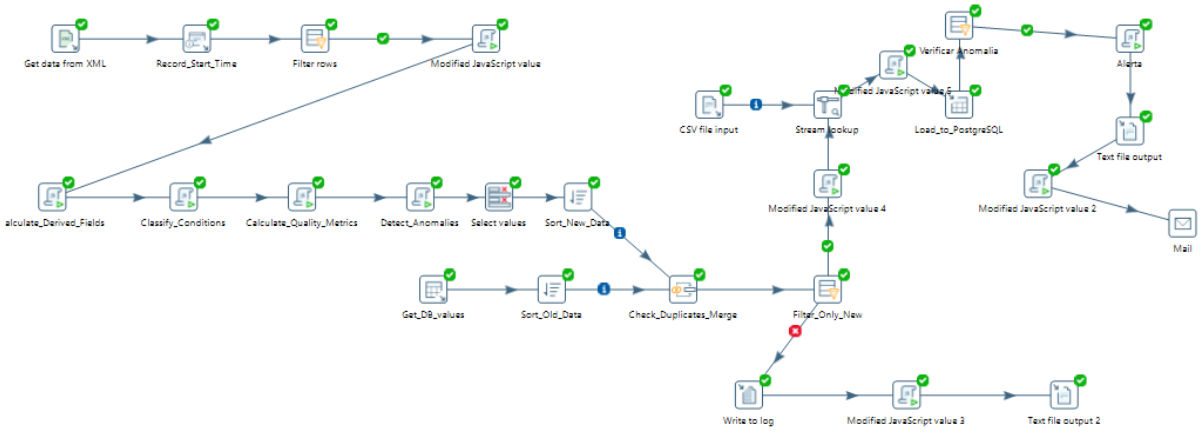

Figura 4: Trans_02_Clean_Transform_Final.ktr

princípios de engenharia de sistemas de informação e com os critérios de mais-valia definidos no enunciado.

5 Jobs

5.1 `Job_Marine_ETL.kjb` – Orquestração do Processo ETL

O *Job* `Job_Marine_ETL.kjb` constitui o ponto de entrada e o orquestrador central de todo o processo ETL. A sua principal função é coordenar, de forma controlada e sequencial, a execução das transformações desenvolvidas (`TR_Carregar_Praias.ktr`, `Trans_01_Extract_Marine_API.ktr` e `Trans_02_Clean_Transform_Final.ktr`). Este mecanismo de controlo reflete a **orquestração de processos de transformação desenvolvidos** e a capacidade de **desenvolver Jobs completos**, de acordo com os critérios de mais-valia definidos no enunciado (Instituto Politécnico do Cávado e do Ave, 2025).

Passos Principais:

1. **Início e Preparação do Ambiente:** O processo tem início no passo *Start*, que marca o início lógico do fluxo ETL. Em seguida, é executado o passo *Check DB Connections*, responsável por verificar a conectividade com a base de dados `Marine_PostgreSQL_DW`. Esta verificação prévia assegura que o ambiente se encontra operacional antes da execução das transformações, evitando falhas por dependências externas. A existência desta validação representa uma boa prática de **gestão de dependências e integridade do sistema**, bem como um exemplo de **operações essenciais sobre bases de dados**.

2. **Execução da Transformação de Praias (`TR_Carregar_Praias.ktr`):** Após a validação do ambiente, é executada a transformação *beachs (Transformation)*, responsável pelo carregamento e validação dos dados das praias a partir do ficheiro `Praias.csv`. Este passo é fundamental para preparar os dados de referência geográfica e estrutural que servirão de base às fases seguintes.

3. **Extração de Dados Marinhos (`Trans_01_Extract_Marine_API.ktr`):** Após a conclusão bem-sucedida do carregamento das praias, é invocada a transformação *load_beachs (Transformation)*. Esta etapa extrai dados oceanográficos em tempo real a partir da API externa `marine-api.open-meteo.com`, recorrendo ao ficheiro `praias_validadas.csv` como referência. Esta etapa exemplifica a integração de **serviços remotos (APIs REST)** no contexto ETL, reforçando o caráter distribuído e dinâmico do sistema.

Figura 5: Job

4. **Transformação e Carga Final (`Trans_02_Clean_Transform_Final.ktr`):** A terceira transformação, *upload (Transformation)*, é responsável pela limpeza, enriquecimento, deteção de anomalias e carregamento final dos dados processados na base de dados analítica `marine_esposende_dw`. Este passo consolida a integração entre dados validados localmente e dados recolhidos remotamente, cumprindo o objetivo de gerar informação limpa e confiável para análise.

5. **Encerramento do Processo:** Após a execução das três transformações, o *Job* culmina no passo *Success*, que confirma a conclusão bem-sucedida do fluxo ETL. Caso ocorra qualquer erro durante a execução, o controlo é transferido para um passo alternativo (*Abort* ou *Error Handling*), que regista o erro num ficheiro de log e envia notificação automática por e-mail, demonstrando a preocupação com a **tolerância a falhas e recuperação de processos**.

O *Job* `Job_Marine_ETL.kjb` garante a execução ordenada das transformações, assegurando a coerência dos fluxos de dados e a integridade do repositório analítico final. O desenho modular do *Job* permite uma fácil manutenção e escalabilidade, possibilitando a adição de novas transformações ou integrações futuras. Além disso, a inclusão de verificações de conectividade, logging e notificações automáticas reflete a adoção de práticas robustas de **governança de dados e monitorização de processos ETL**.

6 Modelo de Dados

O modelo de dados concebido para o *Data Warehouse* `marine_esposende_dw` tem como objetivo suportar a integração, armazenamento e análise de dados oceanográficos e costeiros da região de Esposende. A estrutura segue uma abordagem *star schema*, composta por uma tabela de factos principal, uma dimensão geográfica e uma tabela auxiliar de auditoria, assegurando coerência e rastreabilidade ao longo de todo o processo ETL.

6.1 Descrição das Entidades

Tabela `marine.fact_beachs` Esta tabela contém os metadados das praias, funcionando como dimensão geográfica. Cada registo corresponde a uma praia identificada por um identificador único (`id`), com os atributos `beach_name`, `lat`, `lon` e `orientacao`. Serve de base para associar as medições oceanográficas armazenadas na tabela de factos.

Tabela `marine.fact_marine_data` Tabela de factos principal, onde são armazenadas as medições recolhidas e processadas pela pipeline ETL. Inclui as variáveis físicas (`wave_height`, `sea_surface_temperature`, `ocean_current_velocity`), métricas derivadas (`wave_energy_kj_m2`), classificações (`wave_condition`, `water_temp_class`), e indicadores de qualidade e anomalias (`data_quality_score`, `is_anomaly`). Cada registo é identificado por `id` e associado à praia correspondente através do campo `id_praia`.

Tabela `log.log_table` Tabela auxiliar responsável por registar a execução dos processos ETL desenvolvidos no Pentaho. Armazena dados como o nome do *job*, estado de execução, número de linhas processadas, erros e datas de início e fim, cumprindo o critério de **Geração de Logs e auditoria** do sistema.

6.2 Relações e Estrutura Relacional

O modelo implementa relações diretas entre as tabelas, de modo a garantir a integridade referencial e a rastreabilidade de dados:

- Relação 1:N entre `marine.fact_beachs` e `marine.fact_marine_data`, através de `id_praia`.
- Associação lógica entre `log.log_table` e os processos ETL que geram os dados.

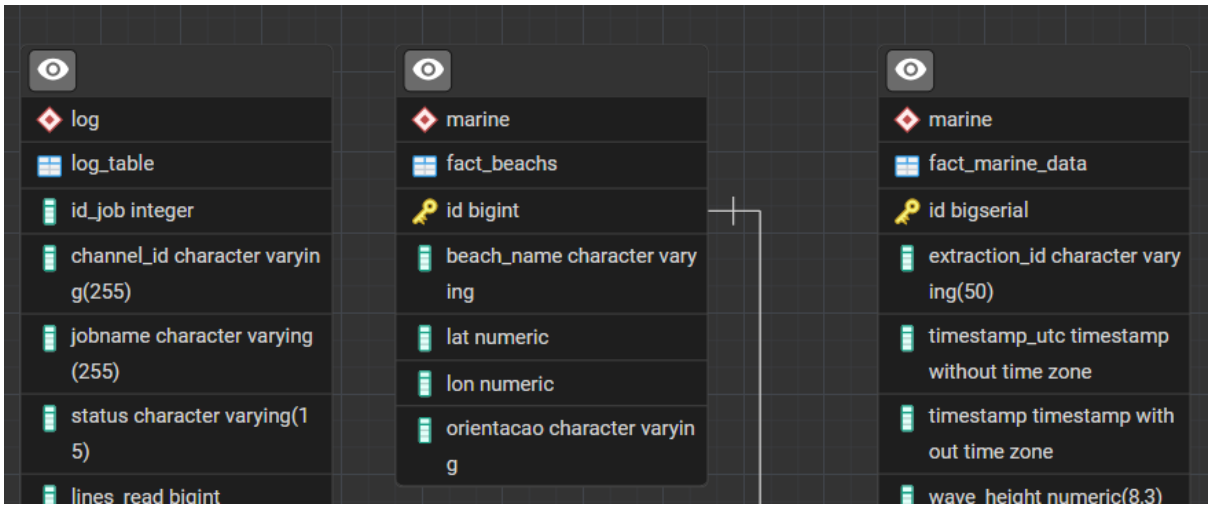

Figura 6: Modelo Entidade-Relacionamento do Data Warehouse marine_esposende_dw.

A Figura apresenta o diagrama Entidade-Relacionamento que representa graficamente estas ligações.

6.3 Resumo das Tabelas

Tabela 1: Resumo das principais tabelas do `marine_esposende_dw`

Tabela	Descrição
`marine.fact_beachs`	Contém os metadados das praias (nome, orientação, latitude e longitude).
`marine.fact_marine_data`	Armazena as medições oceanográficas processadas, métricas derivadas e indicadores de qualidade.
`log.log_table`	Regista a execução dos jobs ETL, incluindo o estado, erros e número de linhas processadas.

6.4 Considerações Finais

Este modelo de dados assegura a integridade, a consistência e a rastreabilidade das informações processadas, garantindo uma base sólida para análises posteriores e visualização de dados em *dashboards*. De acordo com as boas práticas de *Data Warehousing* (Kimball & Ross, 2013), a estrutura proposta privilegia a simplicidade, a clareza e a eficiência no acesso a dados históricos e agregados.

7 Visualização dos Resultados

Para demonstrar a eficácia do processo ETL e a utilidade dos dados integrados, foi desenvolvida uma *Dashboard* interativa. Esta plataforma permite uma visualização clara e sintética das condições oceanográficas e costeiras, facilitando a análise e a monitorização em tempo real. A sua criação cumpre diretamente o critério de **visualização dos resultados conseguidos utilizando dashboards**.

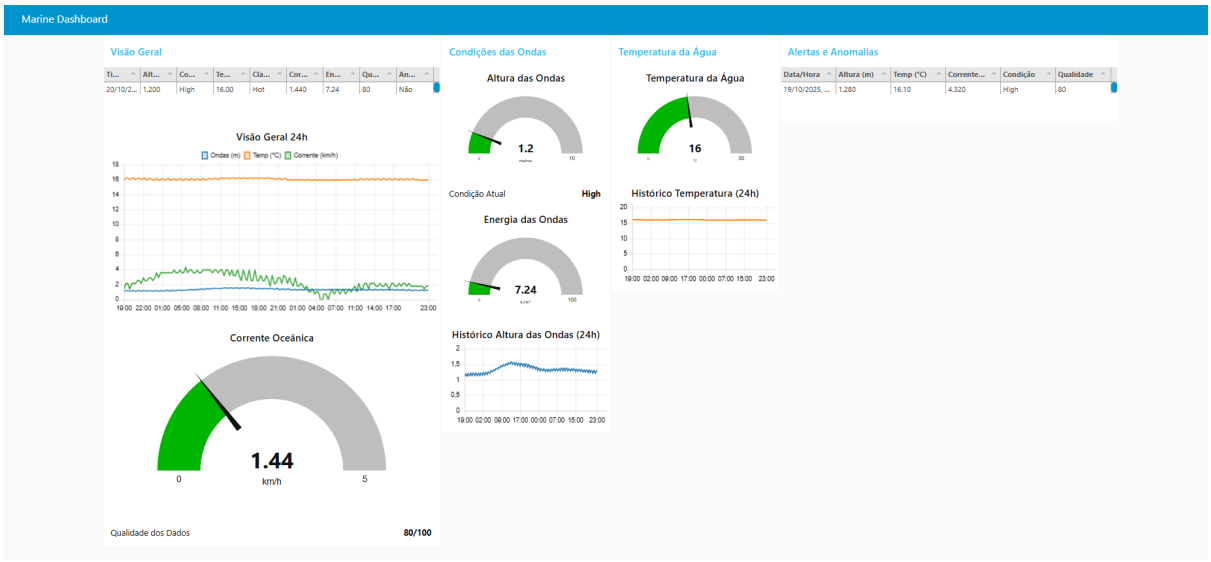

Figura 7: Dashboard interativa para visualização de dados oceanográficos e de praias.

A *Dashboard* apresenta uma visão geral das condições atuais, incluindo altura das ondas, temperatura da água, velocidade das correntes oceânicas e energia das ondas, através de indicadores visuais dinâmicos. Adicionalmente, integra gráficos de séries temporais referentes às últimas 24 horas, permitindo observar tendências e variações significativas.

Uma secção dedicada a **Alertas e Anomalias** destaca automaticamente eventos críticos, como valores extremos de altura das ondas, temperatura da água e intensidade das correntes, complementada por um *score* de qualidade dos dados. Esta visualização constitui uma ferramenta essencial de apoio à decisão e comunicação técnica, permitindo a identificação rápida de padrões e anomalias nos dados oceanográficos recolhidos.

8 Demonstração em Vídeo

Para complementar a documentação técnica e demonstrar a execução prática do processo ETL, foi produzido um vídeo explicativo do projeto. O vídeo apresenta a implementação das transformações no *Pentaho Data Integration*, o fluxo de dados entre as fontes e o *data warehouse*, e a análise interativa realizada na *Dashboard* desenvolvida em Node-RED.

Demonstração do Projeto em Vídeo

Aceder ao vídeo da demonstração no YouTube

O vídeo tem uma duração aproximada de 10 minutos e demonstra, de forma prática, a aplicação dos conceitos de integração, transformação e visualização de dados estudados na unidade curricular de *Integração de Sistemas de Informação*. A demonstração inclui as principais transformações, a execução do *Job* de orquestração e a análise dos resultados obtidos na *Dashboard* de monitorização marítima.

Referência bibliográfica (formato APA):

Santos, G. (2025). *Demonstração do Processo ETL e Dashboard Oceanográfica* [Vídeo]. YouTube. Disponível em: https://youtu.be/Ayt6tMrhYLE

9 Conclusão e Trabalhos Futuros

Este trabalho prático demonstrou com sucesso a implementação de um processo *Extract, Transform, Load (ETL)* completo, robusto e escalável, recorrendo ao *Pentaho Data Integration* para integrar dados de praias e dados oceanográficos numa base de dados *PostgreSQL*. As transformações desenvolvidas abordaram diversos desafios técnicos e ilustraram a aplicação dos principais critérios de mais-valia definidos no enunciado.

- **Diversidade de Operadores de Transformação e Operações sobre Valores:** Foram implementadas operações como o cálculo de métricas derivadas (e.g., energia das ondas), classificações de condições (e.g., `wave_condition`, `water_temp_class`) e deteção de anomalias, evidenciando a **diversidade de operadores de transformação** e a execução de **operações sobre valores** complexas.

- **Utilização de Expressões Regulares (ER):** A validação de nomes de praias recorreu à **utilização de expressões regulares**, assegurando a qualidade e consistência dos dados de entrada.

- **Importação e Exportação de Dados (XML, JSON, CSV):** O projeto demonstrou a capacidade de manipular múltiplos formatos de dados — **XML** (para *staging*), **JSON** (para consumo de API) e **CSV** (para dados locais) — cobrindo uma ampla gama de representações.

- **Desenvolvimento de *Jobs* e Controlo de Processos:** O `Job_Marine_ETL.kjb` atuou como orquestrador central, garantindo a correta execução sequencial das transformações e a sua dependência lógica.

- **Acesso a APIs e Serviços Remotos:** O consumo da API `marine-api.open-meteo.com` exemplificou o **acesso a APIs remotas**, enquanto o envio de alertas por email validou a integração com **serviços remotos**.

- **Joins, Agrupamentos e Lookups:** Foram aplicadas técnicas de **Joins** e **Merging** para evitar duplicação de registos, e **Lookups** para enriquecer os dados oceanográficos com informação complementar das praias.

- **Normalização e Validação de Dados:** Implementou-se a **normalização** (e.g., capitalização e formatação de nomes) e a **validação de qualidade**, garantindo coerência e fiabilidade.

- **Geração de Logs:** Foram desenvolvidos mecanismos robustos de **Geração de Logs** para monitorizar a execução e registar erros e anomalias, assegurando a rastreabilidade do processo.

- **Operações sobre Bases de Dados:** Todas as fases do ETL envolveram **operações essenciais sobre bases de dados**, desde a verificação de conectividade até ao carregamento final das tabelas `marine.fact_beachs` e `marine.fact_marine_data`.

- **Visualização de Resultados:** A criação de uma *dashboard* interativa para visualização dos dados oceanográficos e costeiros constituiu uma mais-valia significativa, demonstrando a **capacidade de transformação de dados em informação acionável**.

A arquitetura proposta, baseada num *job* central que orquestra transformações modulares, assegura uma gestão eficiente do fluxo de dados e a incorporação de lógicas de negócio complexas, como a deteção automática de anomalias e o envio de alertas por email. O uso de uma base de dados relacional como *data warehouse* garante persistência, rastreabilidade e suporte a análises analíticas avançadas.

9.1 Trabalhos Futuros

Para evolução do projeto, diversas direções de **Trabalhos Futuros** são recomendadas. Uma prioridade seria o desenvolvimento de **dashboards interativos** com ferramentas de *Business Intelligence* (e.g., Power BI, Tableau ou Grafana), proporcionando *insights* em tempo real e fortalecendo o critério de **visualização dos resultados conseguidos utilizando dashboards**. Outra vertente essencial é a **otimização de desempenho**, explorando a paralelização de passos, técnicas de *caching* e ajuste de consultas SQL no Pentaho. A **integração com orquestradores avançados** como o Apache Airflow poderá melhorar a escalabilidade e gestão de dependências. A expansão das fontes de dados — incluindo APIs meteorológicas e dados de tráfego marítimo — permitirá análises mais abrangentes e precisas. Adicionalmente, a **aplicação de modelos**

de *machine learning* poderá substituir as regras fixas atuais na deteção de anomalias, proporcionando maior adaptabilidade. Finalmente, a **gestão de metadados** deverá ser formalizada através de um sistema dedicado, reforçando a governança e a rastreabilidade do processo de integração de dados.

Referências Bibliográficas

Apache Software Foundation. (2025). *Apache Airflow Documentation* [[Ferramenta de orquestração de workflows para pipelines de dados]]. https://airflow.apache.org/docs/

Batini, C., & Scannapieco, M. (2009). Data Quality: Concepts, Methodologies and Techniques. *Data & Knowledge Engineering*, *68*(3), 389–398. https://doi.org/10.1016/j.datak.2008.09.002

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 93–104. https://doi.org/10.1145/342009.335388

Grafana Labs. (2025). *Grafana Documentation: Visualize and Explore Metrics* [[Plataforma de visualização e monitorização de dados]]. https://grafana.com/docs/

Inmon, W. H. (2005). *Building the Data Warehouse* (4th). Wiley.

Instituto Politécnico do Cávado e do Ave. (2025). *Enunciado do Trabalho Prático 01 — Integração de Sistemas de Informação* [[Licenciatura em Engenharia de Sistemas Informáticos]].

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd). Wiley.

Microsoft Corporation. (2025). *Microsoft Power BI Documentation* [[Documentação oficial do Microsoft Power BI]]. https://learn.microsoft.com/power-bi/

OpenJS Foundation. (2025). *Node-RED Dashboard User Guide* [[Documentação oficial da framework Node-RED]]. https://nodered.org/docs/ui

Open-Meteo. (2025). *Marine Weather API Documentation* [[API utilizada para extração de dados oceanográficos]]. https://open-meteo.com/en/docs/marine-weather-api

PostgreSQL Global Development Group. (2025). *PostgreSQL 16.3 Documentation* [[Documentação oficial do PostgreSQL]]. https://www.postgresql.org/docs/

Salesforce. (2025). *Tableau Desktop and Web Authoring Help* [[Documentação oficial do Tableau para visualização de dados]]. https://help.tableau.com/

Vantara, H. (2024). *Pentaho Data Integration (Kettle) User Guide* [[Documentação oficial do Pentaho Data Integration]]. https://help.hitachivantara.com/Documentation/Pentaho/9.3/Products/Data_Integration