IPCA INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

Relatório 1ª Fase Trabalho Prático

EST-IPCA

Licenciatura Engenharia de Sistemas Informáticos (Pós-Laboral)

Integração de Sistemas de Informação (ISI)

Docente: Luis Gonzaga Martins Ferreira

Diogo Graça
nº 28004

1º Semestre 2025/2026

Data: 17/10/2025

Indice

Indice de Imagens

1-Introdução

Motivação

Trabalho Prático realizado na Unidade Curricular (UC) *Integração de Sistemas de Informação,* integrada no 1º semestre do 3º Ano com o objetivo de aplicar e consolidar os conceitos associados aos processos ETL (Extract, Transform and Load). O projeto foca-se na utilização de ferramentas e técnicas que permitem integrar, transformar e visualizar dados provenientes de diferentes fontes, nomeadamente ficheiros CSV e serviços web (APIs).

Para a sua execução, foi escolhida a ferramenta KNIME Analytics Platform, pela sua versatilidade na construção visual de workflows ETL, bem como a Google Custom Search API para enriquecimento de dados. A fase final do trabalho inclui o desenvolvimento de um dashboard interativo, que foi realizado em HTML/CSS/JavaScript para permite a visualização e análise dos resultados obtidos.

Problema Proposto

O problema identificado consistia na falta de uma solução integrada para gerir e analisar os dados relativos às reparações de equipamentos informáticos.

Os registos existentes em Excel careciam de normalização, de integração com outras fontes (por exemplo, informações do modelo via web ou o preço dos mesmos) e de ferramentas que permitissem visualizar estatísticas e relatórios de forma visual e dinâmica.

O projeto propõe resolver este problema através de um processo ETL automatizado, consctruído no KNIME, que permitirá:

- Importar e tratar dados de um ficheiro CSV original;
- Integrar informação adicional obtida de uma API externa;
- Exportar os resultados em múltiplos formatos (CSV, JSON, XML);
- Carregar esses dados num dashboard interativo web, desenvolvido para este fim.

2 – Trabalho Desenvolvido

Estratégia Utilizada

O trabalho foi dividido em duas fases complementares:

Fase 1 - ETL no KNIME

- Extração: leitura do ficheiro CSV com os dados originais.
- Transformação: normalização e limpeza de dados (tratamento de valores nulos, formatação de texto, uniformização de booleanos e datas).
- Integração: enriquecimento dos dados através da Google Custom Search API, obtendo informações adicionais como imagem, descrição e link do modelo.
- Carga (Load): exportação dos resultados para ficheiros CSV, JSON e XML.

Fase 2 – Dashboard Web

- Implementação de um dashboard interativo em HTML, CSS e Javascript
- Utilização da biblioteca PapaParse para ler ficheiros CSV diretamente no navegador.
- Geração automática de estatísticas, relatórios e indicadores gráficos.
- Implementação de filtros, pesquisa, e modal de detalhes por r eparação.

Transformações

As principais transformações aplicadas no KNIME foram:

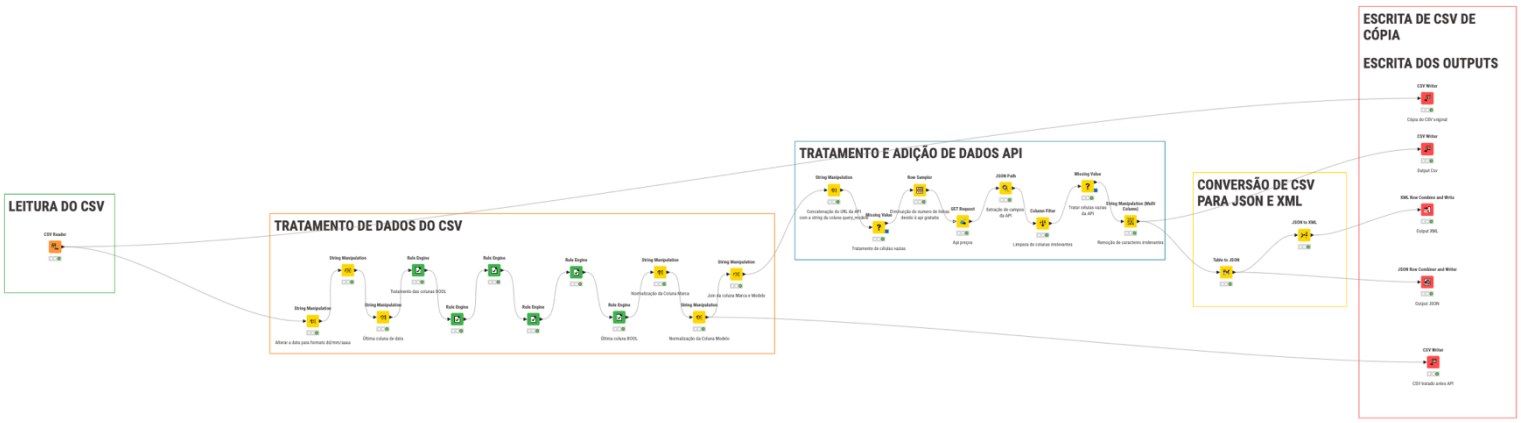

Figura 1- KNIME Workflow

Tipo de Operação	Descrição	Ferramenta KNIME
Normalização de texto	Conversão de nomes, marcas e modelos para formato uniforme	*String Manipulation*
Tratamento de valores em falta	Substituição por “Desconhecido” ou “Sem Observações”	*Missing Value*
Conversão de booleanos	Transformação de “Sim/Não” em valores booleanos	*Rule Engine*
Integração com API	Construção dinâmica de URLs e obtenção de dados remotos	*GET Request*
Extração de dados JSON	Extração de campos relevantes da resposta	*JSON Path*
Exportação de dados	Gravação em CSV, JSON e XML	*CSV Writer, JSON Writer, XML Writer*

Nota: Foi utilizado um “Row Sampler” de modo a melhor visualização dos resultados e para que o numero de requests feitos à API sejam menores uma vez que é gratuita e tem um limite de consultas

Tratamento utilizando Regex

A fase de Transformação (T) do pipeline ETL foi uma das mais relevantes do projeto, uma vez que os dados provenientes do ficheiro CSV original apresentavam inconsistências, formatos díspares e ruído textual.

Recorrendo à ferramenta KNIME Analytics Platform, foram aplicadas várias operações de limpeza, normalização e conversão, garantindo que toda a informação seguisse um formato padronizado e compatível com as fases seguintes de integração e visualização.

Um dos casos mais críticos identificados foi o campo "Data Entrada", que apresentava múltiplos formatos, como ddMMyyyy, yyyy-MM-dd, dd-MM-yyyy e variações com espaços ou separadores incorretos. Para resolver este problema, foi utilizado o nó String Manipulation, explorando o potencial das expressões regulares (regex).

A função composta de regexReplace() foi utilizada de forma encadeada para realizar diferentes substituições e transformações sobre o mesmo campo. Primeiramente, substituíram-se os traços e espaços por barras, uniformizando o separador entre os componentes da data. Em seguida, foram definidos padrões de substituição capazes de reconhecer e converter os diferentes formatos para o padrão final dd/MM/yyyy. Por fim, aplicou-se uma limpeza adicional que removeu quaisquer caracteres não numéricos ou inválidos.

O código final da expressão, aplicado sobre a coluna $Data Entrada$, foi o seguinte:

Expression

```
regexReplace(
    regexReplace(
        regexReplace(
            regexReplace(
                regexReplace(
                    strip($Data Entrada$),
                    "[-\\s]", "/"),     // Troca "-" e espaço por "/"
                "^(\\d{2})(\\d{2})(\\d{4})$", "$1/$2/$3"), // ddMMyyyy para dd/MM/yyyy
            "^(\\d{4})/(\\d{2})/(\\d{2})$", "$3/$2/$1"),   // yyyy/MM/dd para dd/MM/yyyy
        "^(\\d{2})/(\\d{2})/(\\d{4})$", "$1/$2/$3"),       // redundante para garantir formato final
    "[^0-9/]", "")                                        // Remove ruído
```

Figura 2 - Regex Datas

Esta sequência garantiu que qualquer valor, independentemente do formato original, fosse convertido para o formato “dia/mês/ano”, consistente e adequado para ordenação, filtragem e posterior exportação.

A aplicação das expressões regulares mostrou-se uma ferramenta poderosa e flexível, permitindo resolver de forma elegante um problema recorrente em ficheiros CSV provenientes de origens distintas.

Outro aspeto importante do processo de transformação prendeu-se com a conversão de valores textuais em booleanos, particularmente nas colunas “Carregador”, “Bateria” e “Cabo A/C”. Estas colunas, que indicavam a presença ou ausência de acessórios, continham valores muito variados, como “sim”, “s”, “yes”, “1”, “não”, “n”, “no”, “0”, “?” ou mesmo campos vazios.

Para unificar esta informação e permitir o seu tratamento lógico, foi utilizado o nó Rule Engine, que possibilitou definir condições explícitas de correspondência e atribuir os valores lógicos TRUE ou FALSE conforme o caso.

O conjunto de regras aplicadas no campo $Carregador$ foi o seguinte:

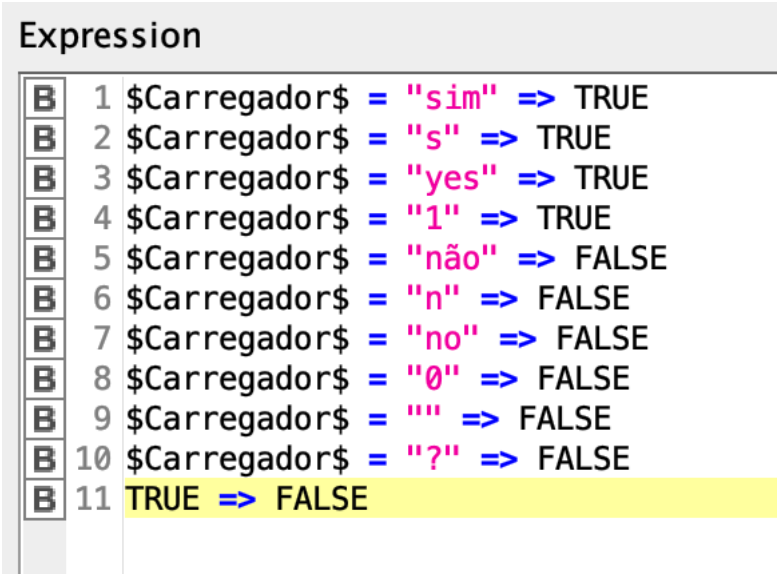

Figura 3 - Rule engine BOOLEANOS

Estas regras foram replicadas para os restantes campos binários, garantindo uma coerência global em toda a base de dados. O uso do Rule Engine permitiu transformar dados ambíguos em valores booleanos precisos e uniformes, facilitando tanto a análise no dashboard web como a exportação para formatos estruturados como JSON e XML.

Além disso, esta conversão simplificou a implementação de indicadores no dashboard, como o número total de reparações aprovadas, prontas ou enviadas, uma vez que os estados podiam ser avaliados diretamente a partir de valores lógicos.

Para além destas transformações principais, foram também realizadas operações complementares, como a remoção de espaços e caracteres especiais em campos textuais (nomes, observações e diagnósticos), a concatenação das colunas “Marca” e “Modelo” para facilitar a pesquisa na API, e a filtragem de colunas com o nó Column Filter, assegurando que apenas os campos relevantes fossem mantidos na exportação final.

Estas etapas, embora mais simples, foram fundamentais para garantir a consistência e qualidade dos dados.

Em síntese, o processo de transformação desenvolvido no KNIME foi responsável por converter dados heterogéneos e potencialmente incorretos em informação estruturada, limpa e coerente. O uso de expressões regulares e regras condicionais representou um elemento distintivo do projeto, revelando a importância da automatização e da padronização na integração de sistemas de informação.

Jobs Desenvolvidos

O workflow completo foi desenvolvido no KNIME Analytics Platform, estruturado num único job principal responsável pelo processo ETL. Este job inclui tarefas de leitura, transformação e integração de dados, bem como subcomponentes para comunicação com a API e exportação final. O diagrama apresentado na Figura 1 ilustra a organização global do pipeline e as relações entre os nós principais descritos nas secções anteriores.

Dashboard e visualização

Após a fase de processamento e exportação no KNIME, os dados resultantes foram utilizados para alimentar uma aplicação web desenvolvida em HTML, CSS e JavaScript, que funciona como um dashboard interativo para análise e consulta das reparações.

Esta componente web representa a fase final do ciclo de integração, onde a informação limpa e estruturada é disponibilizada ao utilizador de forma visual, acessível e intuitiva.

O dashboard foi implementado com três ficheiros principais:

- index.html — estrutura e conteúdo da aplicação;
- style.css — design visual, cores, layout responsivo e componentes de interface;
- script.js — lógica de negócio, manipulação de dados e interatividade.

A interface foi desenvolvida para oferecer uma experiência de utilização fluida e moderna, inspirada em sistemas reais de gestão.

O utilizador pode carregar diretamente o ficheiro CSV exportado do KNIME através de um botão na aba “Importar CSV”. A leitura do ficheiro é feita localmente no navegador, com recurso à biblioteca PapaParse, que converte o conteúdo em objetos JavaScript.

A partir desse momento, toda a aplicação se torna interativa e dinâmica, sem necessidade de ligação a um servidor.

Dashboard

Na aba principal, “Dashboard”, são apresentados indicadores agregados, calculados em tempo real a partir dos dados importados. Estes incluem o total de reparações, o número de reparações aprovadas, prontas e enviadas, apresentados sob a forma de cartões coloridos.

O sistema também mostra as reparações mais recentes numa tabela destacada, permitindo ao utilizador ter uma visão imediata do estado global das operações.

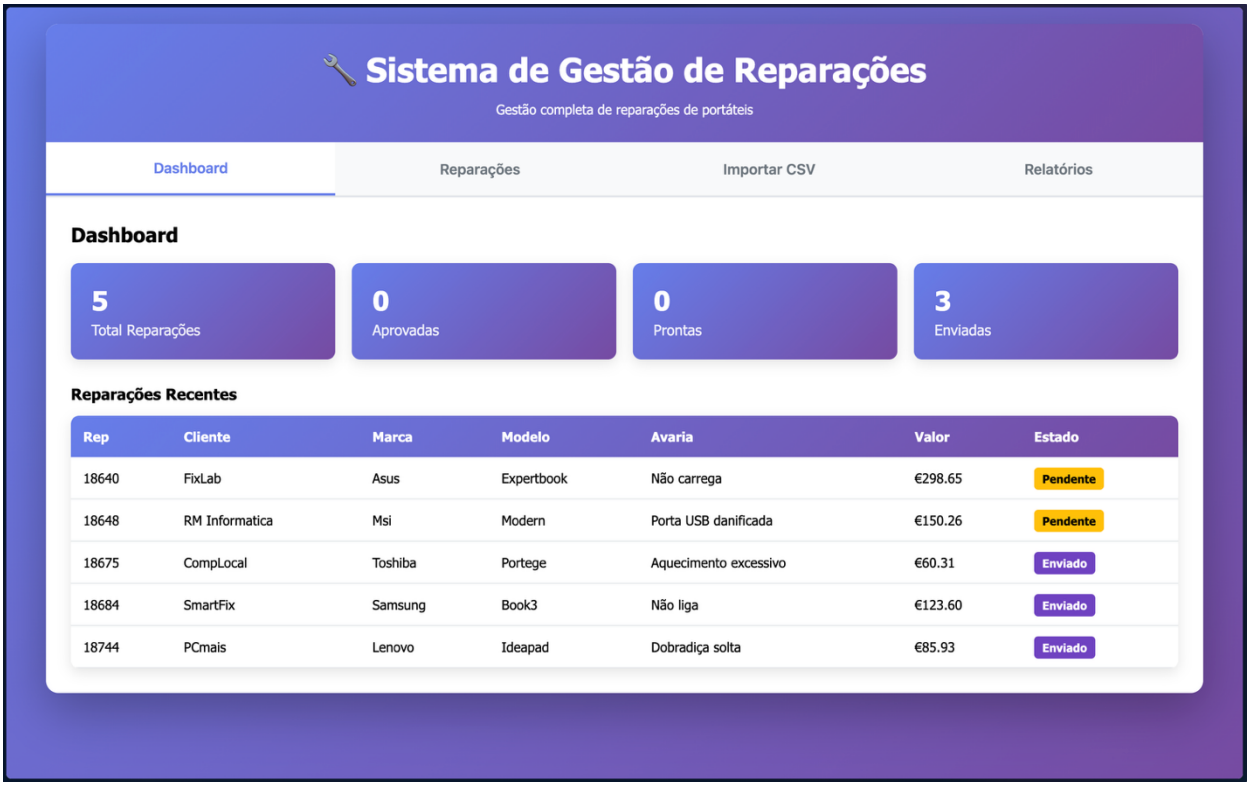

Figura 4 - Dashboard

Reparações

Na aba “Reparações”, é disponibilizada uma tabela completa com todos os registos processados, acompanhada de um sistema de pesquisa e filtros.

O utilizador pode procurar por nome de cliente, marca ou modelo, e filtrar as reparações por estado — aprovado, pronto, enviado ou pendente.

A função applyFilters() em JavaScript é responsável por essa filtragem, e utiliza expressões condicionais para comparar o texto de pesquisa com o estado atual de cada reparação.

Cada linha da tabela inclui ainda um botão “Detalhes”, que abre um popup interativo com todas as informações da reparação, incluindo dados do cliente, acessórios recebidos, diagnósticos, valores e observações.

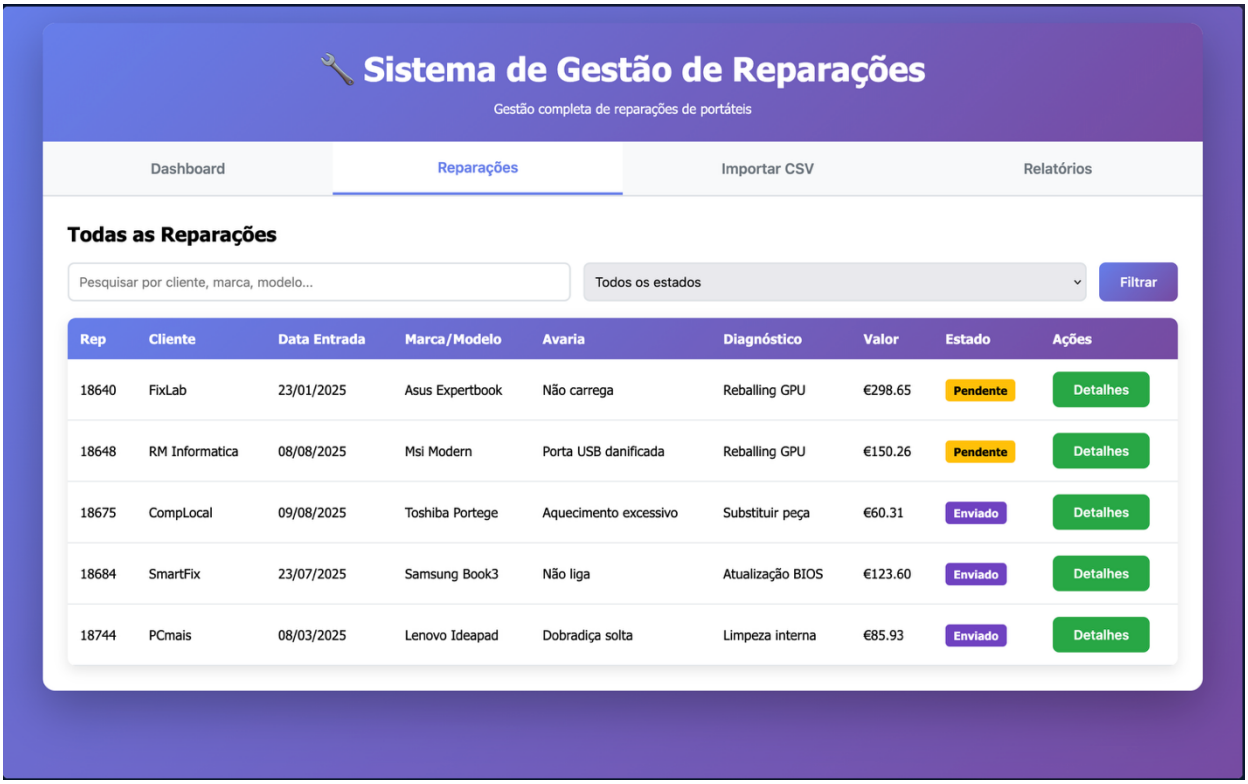

Figura 5 - Reparações

Relatórios

A aba “Relatórios” completa a aplicação, apresentando estatísticas mais detalhadas, como a receita total, o valor médio por reparação, a taxa de conclusão e a taxa de aprovação.

Além disso, é gerado automaticamente um relatório por cliente, que mostra o número de reparações associadas, o valor total faturado e o número de intervenções aprovadas e concluídas.

Todos estes cálculos são realizados localmente através de funções JavaScript que percorrem o array de dados carregado, aplicando agregações simples com métodos como reduce() e filter().

Figura 6 - Relatórios

3 - Video de Demonstração

Link direto

4 – Conclusão

O projeto cumpriu integralmente os objetivos propostos, demonstrando a capacidade de integrar dados de múltiplas fontes, tratá-los e disponibilizá-los de forma estruturada e visual.

O KNIME revelou-se uma ferramenta eficiente para processos ETL, e o dashboard web mostrou como os dados podem ser facilmente interpretados após a sua transformação.

Trabalhos futuros:

- Integração direta com uma base de dados SQL;
- Criação de gráficos dinâmicos (Chart.js / Recharts);
- Ligação do dashboard a uma API REST própria;
- Notificações automáticas de estado (e-mail/SMS).

5 – Bibliografia

KNIME Analytics Platform – https://www.knime.com

Google Custom Search API – https://developers.google.com/custom-search

PapaParse – https://www.papaparse.com