

Relatório do trabalho da disciplina de Integração de Sistemas de Informação (ISI)

# ETL – Integração de dados Clínicos

---

João Pedro Júnior Barbosa – nº27964

Carolina Macedo Branco – nº27983

Licenciatura em Engenharia de Sistemas Informáticos

Outubro de 2025

Afirmo por minha honra que não recebi qualquer apoio não autorizado na realização deste trabalho prático. Afirmo igualmente que não copiei qualquer material de livro, artigo, documento web ou de qualquer outra fonte exceto onde a origem estiver expressamente citada.

João Pedro Júnior Barbosa - nº27964

Carolina Macedo Branco – nº27983

## Índice de Figuras

FIGURA 1 - DIAGRAMA DE FLUXO DO SISTEMA ETL NO KNIME .....	7
FIGURA 2 - COMPONENTES DO WORKFLOW CONSULTAS_EXAMES .....	9
FIGURA 3 - COMPONENTE CONSULTAS.....	10
FIGURA 4 - COMPONENTE EXAMES.....	11
FIGURA 5 - NÓ JOINER.....	12
FIGURA 6 - COMPONENTE EMAIL.....	13
FIGURA 7 - COMPONENTE HASH .....	14
FIGURA 8 - COMPONENTE EXPORT_JSON .....	15
FIGURA 9 - COMPONENTE API_POST .....	16
FIGURA 10 - COMPONENTE DATABASE .....	17
FIGURA 11 - TABELA PACINTE_EXAMES (SQL) .....	18
FIGURA 12 - WORKFLOW EMAIL .....	20
FIGURA 13 - EMAIL ENVIADO .....	21
FIGURA 14 - DIAGRAMA DA ARQUITETURA PENTAHO KETTLE.....	25
FIGURA 15 - TRANSFORMAÇÃO CONSULTAS.....	26
FIGURA 16 - TRANSFORMAÇÃO EXAMES .....	27
FIGURA 17 - TRANSFORMAÇÃO- INTEGRAÇÃO DE DADOS .....	27
FIGURA 18 - TRANSFORMAÇÃO EMAIL .....	28
FIGURA 19 - TRANSFORMAÇÃO EXPORTAR_RESULTADOS .....	29
FIGURA 20 - JOB ETL_MASTER.....	31
FIGURA 21 - RESULTADO DE EXAMES.....	33
FIGURA 22 - RESULTADOS POR TIPO DE EXAME .....	33
FIGURA 23 - CONSULTAS POR ESPECIALIDADE .....	34
FIGURA 24 - VOLUME DE EXAMES POR TIPO .....	34

# Índice

<b>1. ENQUADRAMENTO .....</b>	<b>1</b>
<b>2. PROBLEMA .....</b>	<b>3</b>
2.1. PROBLEMAS A RESOLVER.....	3
2.2. SOLUÇÃO PROPOSTA.....	3
2.3. DECLARAÇÃO SOBRE A ORIGEM DOS DADOS.....	4
<b>3. ESTRATÉGIA UTILIZADA - KNIME .....</b>	<b>5</b>
3.1. IMPORTAÇÃO E EXPORTAÇÃO DE DADOS.....	5
3.2. TRANSFORMAÇÃO DE DADOS .....	5
3.3. ARMAZENAMENTO E DISTRIBUIÇÃO .....	6
3.4. FERRAMENTAS E COMPONENTES .....	6
3.5. ARQUITETURA DO SISTEMA ETL NO KNIME.....	7
<b>4. TRANSFORMAÇÕES - KNIME.....</b>	<b>9</b>
4.1. ESTRUTURA MODULAR E COMPONENTES – CONSULTAS_EXAMES.....	9
4.1.1. Consultas.....	10
4.1.2. Exames.....	11
4.1.3. JOINER.....	12
4.1.4. Email .....	13
4.1.5. Hash .....	14
4.1.6. Export_JSON.....	15
4.1.7. API_post .....	16
4.1.8. DataBase.....	17
<b>5. JOBS – KNIME .....</b>	<b>19</b>
5.1. ARQUITETURA: JUSTIFICAÇÃO DA COMUNICAÇÃO.....	19
5.1.1. Limitações na Ligação Direta (ETL -> Email) .....	19
5.1.2. Solução Ficheiros e Variáveis de Fluxo.....	19
5.2. WORKFLOW EMAIL (ENVIO DE NOTIFICAÇÕES).....	20
5.2.1. Validação do Envio (Mensagem Final) .....	21
<b>6. PENTAHO KETTLE (SPOON) .....</b>	<b>23</b>
6.1. ESTRATÉGIA UTILIZADA - PENTAHO KETTLE (SPOON).....	23
6.1.1. Importação e Exportação de Dados.....	23
6.1.2. Transformação de Dados.....	23
6.1.3. Armazenamento e Distribuição .....	24
6.1.4. Ferramentas e Componentes.....	24
6.1.5. Arquitetura do Sistema ETL no Pentaho Kettle .....	25
6.2. Transformação-Consultas .....	25
6.3. TRANSFORMAÇÃO-EXAMES .....	27
6.4. TRANSFORMAÇÃO - INTEGRAÇÃO DOS DADOS (MERGE JOIN) .....	27
6.5. TRANSFORMAÇÃO - EMAIL .....	28
6.6. TRANSFORMAÇÃO - EXPORTAR_RESULTADOS .....	29
6.7. JOB-ETL_MASTER.....	31
6.8. GERAÇÃO E ANÁLISE DOS LOGS DE EXECUÇÃO .....	31

<b>7. DASHBOARD .....</b>	<b>33</b>
7.1. RESULTADOS DE EXAMES .....	33
7.2. RESULTADOS POR TIPO DE EXAME.....	33
7.3. CONSULTAS POR ESPECIALIDADE.....	34
7.4. VOLUME DE EXAMES POR TIPO.....	34
<b>8. VÍDEOS.....</b>	<b>36</b>
<b>9. CONCLUSÃO .....</b>	<b>37</b>
<b>10. BIBLIOGRAFIA .....</b>	<b>39</b>

## Resumo

O presente trabalho tem como objetivo o desenvolvimento de um processo ETL (Extract, Transform, Load) destinado à integração e normalização de dados clínicos provenientes de duas origens distintas: marcações de consultas (ficheiros XML) e resultados laboratoriais (ficheiros CSV). O processo compreende as fases de validação e anonimização de dados pessoais, normalização de formatos e integração num repositório central em Microsoft SQL Server, com vista à disponibilização de dados consolidados para análise e suporte à decisão.

Para além das operações de extração, transformação e armazenamento, foi implementada uma simulação de envio de dados (método POST) para uma API, representando a integração com sistemas externos de informação em saúde.

O projeto foi desenvolvido de forma colaborativa recorrendo a duas plataformas de integração de dados: KNIME e Pentaho Kettle (Spoon). Esta abordagem permitiu a execução e comparação de processos ETL em ambientes distintos, com base em fluxos visuais de modelação de dados.

Os resultados obtidos evidenciam a execução correta do processo de integração e a criação de um pipeline funcional capaz de assegurar a coerência e rastreabilidade dos dados tratados.

## 1. Enquadramento

Equipa: João Pedro Júnior Barbosa nº27964, Carolina Macedo Branco nº27983

Unidade Curricular: Integração de Sistemas de Informação (ISI)

Curso: Licenciatura em Engenharia de Sistemas Informáticos

Motivo da Escolha do Tema:

O tema da integração de dados clínicos foi selecionado devido à necessidade de consolidar informação proveniente de sistemas clínicos heterogéneos, com formatos distintos como XML e CSV. A integração permite a normalização e preparação dos dados para análise e suporte à decisão. O desenvolvimento de um processo ETL completo foi realizado para aplicar técnicas de extração, transformação e carregamento de dados, garantindo consistência, anonimização e comunicação com serviços externos.

Plataforma Utilizada por Cada Elemento:

- João Pedro Júnior Barbosa: Pentaho Kettle (Spoon)
- Carolina Macedo Branco: KNIME

Repositórios git:

- <https://github.com/JoaoBarbosaaa/tp01-27964.git>
- <https://github.com/CarolinaMB-22/TP01-27983.git>





## 2. Problema

A integração de dados de saúde constitui um desafio recorrente nos sistemas de informação clínicos, devido à diversidade de origens, formatos e estruturas de dados. A coexistência de sistemas heterogêneos, desenvolvidos de forma independente, provoca fragmentação da informação e limita a capacidade de análise global das atividades clínicas.

O presente trabalho concentra-se na integração de dados provenientes de diferentes fontes, nomeadamente marcações de consultas médicas em formato XML e resultados laboratoriais em formato CSV. O objetivo é garantir a consistência, integridade e anonimização da informação, permitindo a consolidação num repositório central baseado em Microsoft SQL Server e viabilizando consultas uniformizadas e suporte a processos de análise e decisão.

### 2.1. Problemas a Resolver

**Auditoria e Qualidade de Dados:** Desenvolver processos de normalização e validação para limpar os dados brutos, separando registos válidos dos rejeitados (auditorias a dados).

**Integração e Anonimização de Dados:** Unir os dados limpos de Exames e Consultas (joins) e aplicar operações de hashing para anonimizar campos sensíveis (nome, e-mail, NIF).

**Acesso a Serviços Remotos:** Criar um fluxo de trabalho (job) que inclua o acesso a serviços remotos, como envio de e-mail para notificações e submissão de resultados finais em JSON a um serviço web (API) via HTTP POST em formato JSON.

### 2.2. Solução Proposta

Para enfrentar estes problemas, foi desenvolvido um processo ETL (Extract, Transform, Load) automatizado, que inclui:

- Validação de dados com expressões regulares;
- Junção de eventos clínicos por paciente;
- Anonimização de campos sensíveis;
- Transformação de dados para JSON
- Simulação de comunicação com uma API externa via HTTP POST;
- Envio automático de notificações por e-mail sobre novos resultados laboratoriais.

O trabalho foi implementado utilizando duas plataformas de integração de dados, KNIME e Pentaho Kettle (Spoon), permitindo comparar abordagens distintas à implementação de processos ETL, com base em fluxos de transformação visuais e pipelines orientados a componentes.

Seguidamente será apresentada a solução implementada em Pentaho Kettle e KNIME, detalhando as transformações e jobs desenvolvidos.

### **2.3. Declaração sobre a Origem dos Dados**

É importante notar que, dada a natureza restrita dos dados clínicos reais e a ausência de acesso a uma API de dados clínicos válidos para desenvolvimento, o dataset utilizado neste trabalho é sintético.

Os dados foram criados para simular a estrutura de um cenário clínico. O foco do projeto é a demonstração da Arquitetura ETL e a implementação das suas funcionalidades, como a anonimização e a integração de sistemas.

### 3. Estratégia Utilizada - knime

A estratégia adotada para a integração de dados clínicos baseia-se no desenvolvimento de um processo ETL (Extract, Transform, Load) automatizado, estruturado em três fases principais: extração, para o processamento de dados e Jobs para a orquestração do fluxo de trabalho e armazenamento de dados. O processo inclui operadores e componentes específicos para garantir qualidade, consistência e anonimização dos dados.

#### 3.1. Importação e Exportação de Dados

A extração de dados incidiu sobre duas fontes principais: as marcações de consultas médicas, em formato XML, e os resultados laboratoriais, em formato CSV. O processo iniciou-se com a leitura e parsing dos ficheiros estruturados, utilizando operadores adequados para cada formato, e incluiu validações para assegurar a consistência dos dados extraídos. Para os ficheiros XML, recorreu-se ao XML Reader combinado com XPath, permitindo identificar e extrair os campos relevantes das consultas, enquanto os ficheiros CSV foram processados com o CSV Reader, garantindo a leitura correta de todos os registos laboratoriais.

O processo de exportação de dados estruturados dividiu os registos em dois ficheiros distintos. Os registos válidos foram gravados no ficheiro aceites.csv, enquanto os registos inválidos foram separados e gravados no ficheiro rejeitados.csv, permitindo auditoria e controlo da qualidade dos dados. Todas as etapas e resultados de execução foram registados no ficheiro de log etl.log, assegurando rastreabilidade e monitorização do processo ETL. Esta metodologia permitiu uma extração consistente e controlada, preparando os dados para a fase de processamento e integração subsequente.

#### 3.2. Transformação de Dados

A transformação de dados consistiu na validação, normalização, integração e anonimização dos registos clínicos, preparando-os para armazenamento. A validação utilizou expressões regulares para verificar o formato de campos críticos, como datas, NIF e emails, e identificar registos inválidos, que foram gravados no ficheiro rejeitados.csv. Os registos válidos foram armazenados no ficheiro aceites.csv para posterior processamento.

A integração de dados realizou-se através da junção de eventos clínicos por paciente, combinando informações de consultas e exames num único conjunto de dados, garantindo consistência entre as diferentes fontes. Foram também aplicadas operações de normalização de campos textuais, conversão de formatos de data e criação de colunas derivadas para manter uniformidade e facilitar consultas.

A anonimização dos dados *sensíveis*, nomeadamente nome, e-mail e NIF, foi feita mediante hashing SHA-256, preservando a análise estatística sem expor informação pessoal. Todas as operações foram implementadas com componentes da plataforma ETL,

incluindo Column Expressions, String Manipulation, Joiner e Java Snippet, permitindo execução automatizada e rastreável de todo o processo.

### **3.3. Armazenamento e Distribuição**

Após o processamento, os dados transformados foram gravados no repositório central em Microsoft SQL Server, garantindo integridade e consistência. Este repositório permite a realização de consultas uniformizadas e fornece suporte aos processos de análise e decisão.

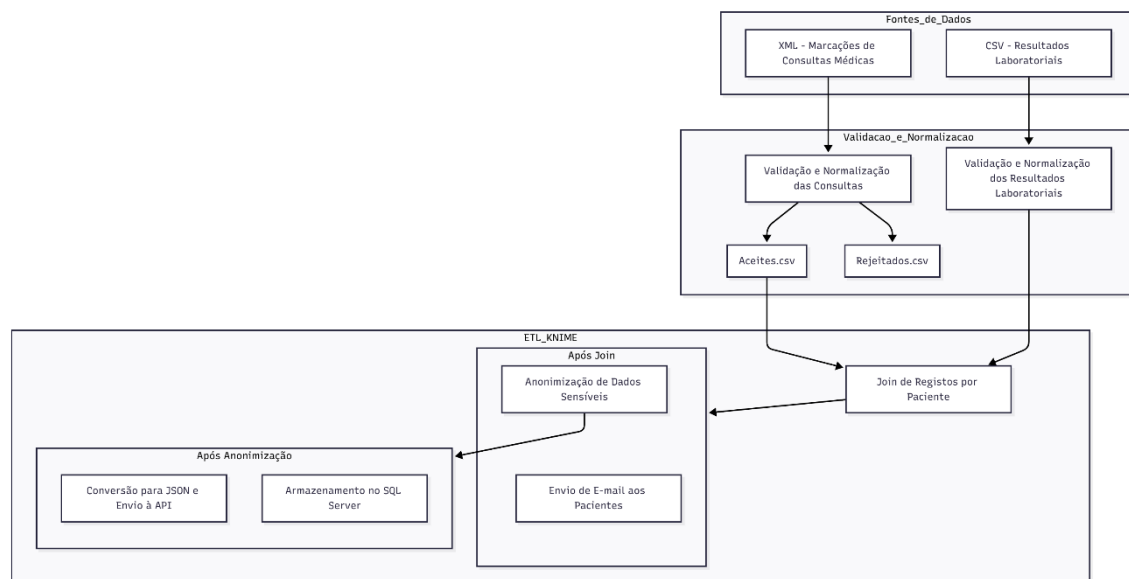
O processo incluiu igualmente a integração com serviços externos, com o envio dos resultados em JSON para uma API simulada através do método HTTP POST e a notificação automática dos pacientes por e-mail sobre novos resultados laboratoriais. Esta abordagem permite disponibilizar os dados transformados para utilização interna e para comunicação com sistemas externos, mantendo rastreabilidade e controlo do fluxo ETL.

### **3.4. Ferramentas e Componentes**

O processo ETL foi implementado em KNIME, utilizando fluxos de transformação visuais para organizar a extração, processamento e armazenamento dos dados clínicos. Foram utilizados operadores para leitura de ficheiros XML e CSV, junção de registos por paciente, normalização de campos, anonimização de dados sensíveis e gravação em ficheiros ou base de dados.

O KNIME permite a execução automatizada e rastreável de todas as etapas do processo, assegurando consistência e integridade dos dados. Esta plataforma facilita a monitorização e auditoria do processo ETL, permitindo reproduzir e controlar todas as operações realizadas.

### 3.5. Arquitetura do Sistema ETL no KNIME



**Figura 1- Diagrama de fluxo do Sistema ETL no KNIME**

O diagrama da figura 1 representa a arquitetura do sistema de integração de dados clínicos implementado em KNIME. As fontes de dados incluem marcações de consultas médicas em formato XML e resultados laboratoriais em formato CSV. Ambos os conjuntos de dados passam por processos de validação e normalização para garantir consistência antes das etapas subsequentes.

As consultas válidas são gravadas no ficheiro aceites.csv, enquanto as inválidas são registadas em rejeitados.csv para auditoria. Os registos válidos das consultas são combinados com os resultados laboratoriais através de um join por paciente, consolidando a informação clínica.

Após o join, o fluxo divide-se em duas operações paralelas: envio de e-mails aos pacientes com resultados disponíveis e anonimização dos dados sensíveis, incluindo nome, e-mail e NIF. Depois da anonimização, os dados seguem paralelamente para conversão em JSON e envio a uma API externa, bem como para armazenamento no Microsoft SQL Server.

A arquitetura modular permite executar todas as etapas de forma automatizada em KNIME, mantendo consistência, rastreabilidade e integridade dos dados clínicos.



## 4. Transformações - knime

Esta secção descreve o conjunto de transformações implementadas no processo ETL (Extract, Transform, Load) desenvolvido em KNIME.

O objetivo é apresentar as operações executadas nas diferentes fases do processo - extração, transformação e carga - e a forma como estas contribuem para a consolidação e integração dos dados clínicos utilizados no projeto.

As transformações incluem a leitura de dados provenientes de múltiplas fontes, a aplicação de regras de validação e normalização, a anonimização de campos sensíveis e o carregamento dos resultados tratados numa base de dados relacional. Cada etapa foi organizada de forma modular, permitindo a execução sequencial e controlada das operações essenciais ao fluxo ETL.

### 4.1. Estrutura Modular e Componentes - consultas\_exames

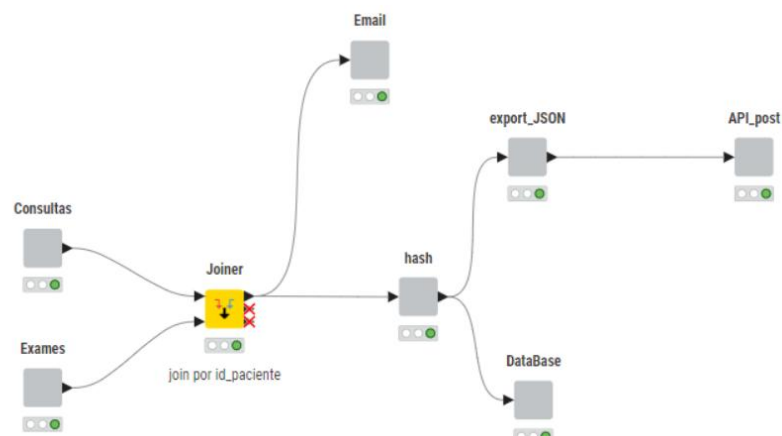


Figura 2 - Componentes do Workflow consultas\_exames

O workflow de ETL consultas\_exames foi concebido com uma arquitetura modular, baseada no conceito de Componentes do KNIME, como se ilustra na *Figura 2*. Esta abordagem otimiza a clareza e a manutenção.

Cada funcionalidade do processo ETL foi isolada num módulo dedicado. A segmentação em Componentes permite o desenvolvimento e a execução independente de cada etapa, facilitando a identificação e correção de erros sem impactar o fluxo global.

As principais Componentes implementadas no workflow são: Consultas, Exames, Joiner, Hash, DataBase, export\_JSON, API\_post e Email.

As secções seguintes detalham o objetivo, os nós e as operações realizadas em cada módulo, em conformidade com as regras de Extração, Transformação e Armazenamento (ETL).

#### 4.1.1. Consultas

Esta componente executa a extração (Extract) e o pré-processamento inicial dos dados de marcações de consultas médicas, cuja fonte primária é o ficheiro estruturado em XML (consultas.xml). O objetivo é preparar e validar os dados, garantindo que estão prontos para a fase subsequente de integração e anonimização.

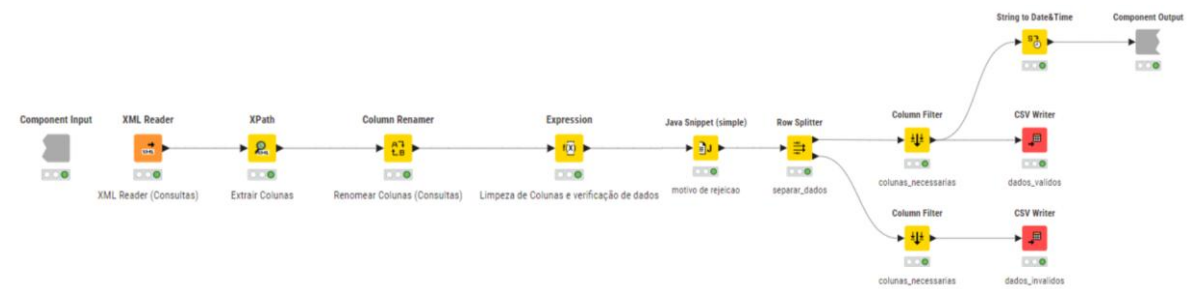


Figura 3 - Componente Consultas

#### Nós utilizados:

- XML Reader (Consultas)
- XPath (Extrair Colunas)
- Column Renamer (Renomear Colunas)
- Expression (Limpeza de Colunas e Verificação de Dados)
- Java Snippet (Simple - motivo de rejeição)
- Row Splitter (separar\_dados)
- Column Filter
- String to Date&Time
- CSV Writer (dados\_validos (aceites) / dados\_invalidos (rejeitados))

#### Descrição do fluxo

O fluxo inicia com o XML Reader, que carrega a estrutura de dados do ficheiro XML. O nó XPath é subsequentemente utilizado para navegar na hierarquia do XML e extrair as colunas chave, como id\_paciente, para a tabela de dados como se pode observar na Figura 3.

A fase de limpeza e normalização ocorre no Column Renamer, para padronização inicial dos nomes das colunas, e no nó Expression, que executa um conjunto de operações de qualidade de dados. A limpeza de espaços é realizada através da função strip() aplicada a campos de texto críticos, removendo espaços em branco nas extremidades dos valores. Para a normalização, aplicam-se replace\_diacritics() na coluna nome, capitalize() em especialidade e lower\_case() em email para uniformização de formatos. A validação dos dados críticos (nif, telemóvel e email) é implementada através de expressões regulares (RegEx), gerando uma flag que sinaliza registos não conformes.

Os registos são então encaminhados para o Java Snippet, que atribui o motivo de rejeição, e o Row Splitter (separar\_dados) divide o fluxo em dois caminhos: registos válidos e inválidos.



No fluxo dos dados válidos, o nó String to Date&Time é aplicado para converter a coluna da data de consulta do formato string para o tipo de dados Date&Time, garantindo a consistência de tipos.

Em ambos os fluxos, o nó Column Filter é utilizado como etapa de preparação final. Este nó seleciona explicitamente apenas as colunas de dados essenciais e descarta colunas auxiliares, otimizando o dataset antes da persistência.

Os dados são finalmente persistidos através de dois CSV Writer separados, identificando claramente os registos dados\_validos (aceites) e dados\_invalidos (rejeitados). O fluxo de dados válidos segue para a próxima componente do workflow principal, o nó Joiner.

### 4.1.2. Exames

Esta componente executa a extração e a pré-transformação dos resultados laboratoriais, utilizando como fonte o ficheiro CSV (exames.csv). O objetivo é estruturar os dados, normalizar os nomes das colunas e garantir a conversão correta dos tipos de dados essenciais antes da fase de integração.



Figura 4 - Componente Exames

#### Nós utilizados:

- CSV Reader (Exames)
- Column Renamer (Renomear Colunas - Exames)
- Expression (Limpeza de Colunas)
- Java Snippet (Simple - datas\_iso)
- String to Date&Time
- Component Output

#### Descrição do fluxo

Como se pode observar na *Figura 4*, o fluxo inicia com o CSV Reader para a leitura dos dados, com a configuração de delimitador e inferência de tipos. Segue-se o Column Renamer (Renomear Colunas - Exames), onde se padronizam os nomes das colunas de acordo com o esquema de destino.

A fase de limpeza de dados é realizada no nó Expression (Limpeza de Colunas), onde se aplicam as seguintes transformações para garantir a qualidade dos dados. A Limpeza

de Espaços foi implementada pela função `strip()` aplicada a todas as colunas chave – `id_paciente`, `tipo_exame`, `resultado` e `data_exame` – removendo espaços em branco iniciais e finais que poderiam comprometer a correspondência (`join`) ou a conversão de tipo. A Normalização de Conteúdo centrou-se na coluna `resultado`, que foi uniformizada através da função `capitalize()`, assegurando que a primeira letra de cada palavra estivesse em maiúscula, mantendo assim a consistência na representação dos resultados laboratoriais.

Em seguida, o Java Snippet (`datas_iso`) é utilizado para forçar a normalização do formato da data. Especificamente, converte-se a string da data de realização do exame do formato de origem (`dd/mm/yyyy`) para o formato padrão ISO (`yyyy-mm-dd`), sendo esta uma etapa de preparação crítica antes da conversão de tipo.

Finalmente, o nó `String to Date&Time` é aplicado para converter a coluna da data de realização do exame do formato string para o tipo de dados `Date&Time` nativo.

O dataset processado e com tipos de dados corrigidos é então emitido através do Component Output e segue como entrada 'Right' para a próxima componente do workflow principal (o nó `Joiner`).

### 4.1.3. JOINER

Este nó representa a primeira fase de integração dos dados, onde a informação das consultas e dos exames é consolidada. A operação de `Join` é crítica para criar o registo unificado do paciente, necessário para as transformações subsequentes e para a carga final na base de dados.



Figura 5 - Nó Joiner

#### Nós utilizados:

- Joiner

#### Descrição do fluxo

O nó `Joiner` recebe dois fluxos de dados de entrada, a tabela de Consultas (fluxo Left) e a tabela de Exames (fluxo Right), e a operação de ligação é configurada para garantir a correspondência precisa entre os registos.

A chave de ligação é efetuada exclusivamente pelo campo `id_paciente`, que atua como chave primária de correspondência entre as duas fontes.

É aplicado um Inner Join o que significa que apenas os registos que possuem correspondência válida do `id_paciente` em ambas as tabelas (Consultas e Exames) são mantidos no fluxo de saída, sendo os registos existentes em apenas uma das fontes descartados nesta fase de integração.

O processo de Join resulta na criação de uma única tabela horizontal, onde cada linha contém agora a informação agregada de consulta e exame para o mesmo paciente, e esta tabela de dados consolidados constitui o dataset de entrada para a próximas componentee, Hash e Email.

#### 4.1.4. Email

Este módulo tem a função crítica de isolar e preparar o dataset necessário para o sistema de notificação por e-mail. Está posicionado no workflow antes da componente hash para aceder aos endereços de e-mail não anonimizados. O módulo persiste os dados filtrados num ficheiro auxiliar, que será processado por um workflow de envio dedicado.

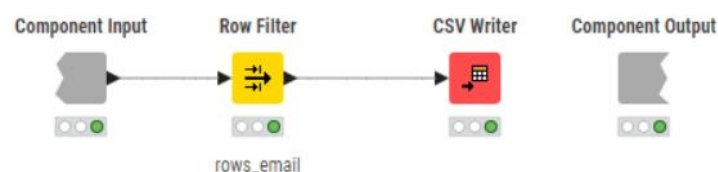


Figura 6 - Componente Email

#### Nós utilizados:

- Row Filter (`rows_email`)
- CSV Writer

#### Descrição do fluxo

Como se pode observar na *Figura 6* o fluxo recebe o dataset consolidado diretamente da saída do nó Joiner, contendo ainda os Dados de Identificação Pessoal (DIP). A operação central é realizada pelo Row Filter (`rows_email`), com o objetivo primário de isolar um subconjunto de registos para fins de teste funcional.

A operação central é realizada pelo Row Filter (`rows_email`), com o objetivo primário de isolar um subconjunto de registos para fins de teste funcional. Embora o dataset utilize e-mails simulados, o nó Row Filter aplica uma expressão regular (Regex) específica para selecionar apenas dois pacientes. Estes dois pacientes foram escolhidos por possuírem

endereços de e-mail simulados que correspondem a contas de teste reais e acessíveis ao utilizador. Esta técnica assegura que, no final do processo, se possa verificar a entrega bem-sucedida da notificação, validando assim a funcionalidade do módulo de envio de e-mail sem comprometer dados reais.

O dataset resultante desta filtragem, contendo os endereços reais de teste e os dados clínicos consolidados, é então escrito através do nó CSV Writer. Este ficheiro CSV atua como a Tabela Auxiliar que contém a lista de notificações pendentes. O Component Output finaliza a componente. A leitura e o processamento para o envio de e-mail a partir deste ficheiro serão detalhados numa secção posterior.

#### 4.1.5. Hash

Esta componente representa a fase final da transformação de dados, focada na segurança e conformidade. O principal objetivo é a anonimização dos dados de identificação pessoal (DIP) através de técnicas criptográficas, garantindo que os dados armazenados na base de dados central sejam pseudonimizados e não diretamente rastreáveis ao indivíduo.



Figura 7 - Componente Hash

#### Nós utilizados

- Java Snippet (passar dados para hash)
- Column Filter (filtrar dados não hash)
- Column Resorter (ordenar colunas)

#### Descrição do fluxo

O fluxo recebe a tabela de dados consolidada proveniente do nó Joiner como de pode observar na *Figura 7*. A primeira operação ocorre no nó Java Snippet (passar dados para hash). Neste snippet de código, é implementada a função criptográfica SHA-256 para gerar um valor de hash.

As colunas de identificação pessoal, como nome, nif e email, são substituídas pelo seu hash correspondente. Esta operação pseudonimiza o dataset, mantendo a integridade dos dados clínicos e de evento, mas eliminando a identificação direta do paciente, cumprindo requisitos de proteção de dados.

Em seguida, o Column Filter (filtrar dados não hash) é aplicado. Este nó remove as colunas de dados de identificação originais após a geração do hash, assegurando que

apenas as versões anonimizadas e os dados clínicos não sensíveis sigam para as fases de armazenamento.

O nó Column Resorter (ordenar colunas) é utilizado para padronizar a ordem das colunas de saída, facilitando a leitura e alinhando-as com o esquema final do banco de dados.

O dataset final, agora anonimizado e estruturado, é emitido através do Component Output. Este fluxo ramifica-se para as fases finais do workflow: armazenamento para a base de dados, exportação JSON e notificação por email.

#### 4.1.6. Export\_JSON

Esta componente assegura a conversão final do dataset ETL para o formato JSON (JavaScript Object Notation), um requisito comum para a partilha de dados com aplicações web ou serviços externos. O objetivo é fornecer uma saída estruturada e serializada dos registos consolidados e anonimizados.



Figura 8 - Componente Export\_JSON

#### Nós utilizados:

- Table to JSON
- JSON Writer

#### Descrição do fluxo

O fluxo recebe o dataset anonimizado e estruturado diretamente da saída da componente hash.

Como se pode observar na *Figura 8*, o processo inicia-se com o nó Table to JSON, que constitui a primeira etapa de conversão. Este nó serializa a tabela de dados, convertendo cada linha (record) do formato tabular KNIME para um objeto JSON. Esta conversão é fundamental para estruturar os dados de forma hierárquica e legível por máquina.

Em seguida, o nó JSON Writer grava o dataset serializado num ficheiro .json. Este ficheiro constitui a saída final do workflow e é destinado a ser utilizado por aplicações de terceiros ou para fins de armazenamento de dados.

O Component Output finaliza o módulo, permitindo a propagação do fluxo de dados (embora geralmente desnecessária após uma operação de gravação final) para a próxima componente na sequência de saída, que é a API\_post (Simulação de API).

#### 4.1.7. API\_post

Esta componente demonstra a capacidade de integração de sistemas do workflow ETL. A sua função é simular o envio dos dados consolidados e anonimizados para um servidor parceiro externo através do método HTTP POST, que é o método padrão para escrever/criar novos registos numa API.

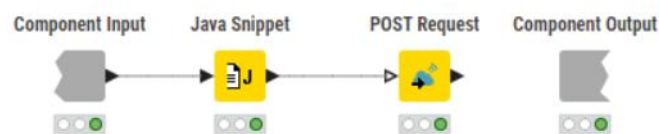


Figura 9 - Componente API\_post

#### Nós utilizados:

- Java Snippet
- POST Request

#### Descrição do fluxo

O fluxo recebe o dataset anonimizado e estruturado, que é a saída final do processamento ETL. Como se pode observar na *Figura 9*, a primeira etapa é executada pelo nó Java Snippet.

Dentro do snippet, a operação chave é a conversão final do dataset para o formato de envio, utilizando a função `out_paciente_eventos = c_paciente_eventos.toString();`. Esta linha assegura que a coluna de dados estruturados (`c_paciente_eventos`) seja explicitamente convertida para o tipo de dados String, o formato exigido para o corpo da requisição HTTP do nó POST Request subsequente.

Em seguida, o nó POST Request é o ponto de comunicação real. Este nó utiliza o URL `https://postman-echo.com/post`. Este endpoint, uma API de teste pública, garante que a configuração técnica (conversão para String JSON, headers e o envio) está correta e funcional. O sucesso da requisição, sinalizado pelo código de *status* HTTP 200 (OK), confirma que a transferência de dados para o sistema externo foi realizada com sucesso.

O uso desta API de teste permite provar a capacidade de submissão de dados para qualquer API real, sem depender de chaves de autenticação (API Keys), e sem correr o

risco de violar a segurança ao usar URLs desconhecidos. O Component Output finaliza o módulo, indicando a conclusão da operação de transferência para o sistema externo.

#### 4.1.8. DataBase

Esta componente é responsável pelo armazenamento final do dataset ETL na base de dados relacional. É o passo conclusivo do processo, onde os dados anonimizados, limpos e estruturados são carregados para o esquema de destino para posterior análise e consumo por aplicações.

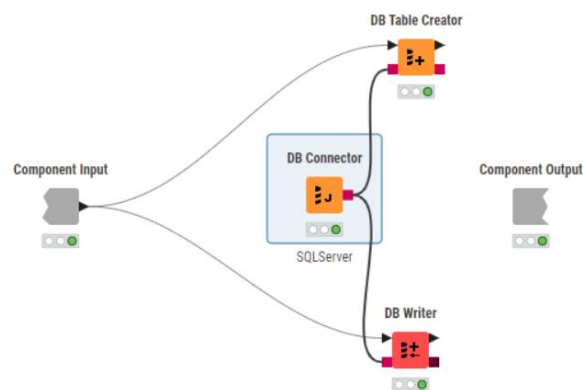


Figura 10 - Componente DataBase

##### Nós utilizados:

- DB Connector (SQLServer)
- DB Table Creator
- DB Writer

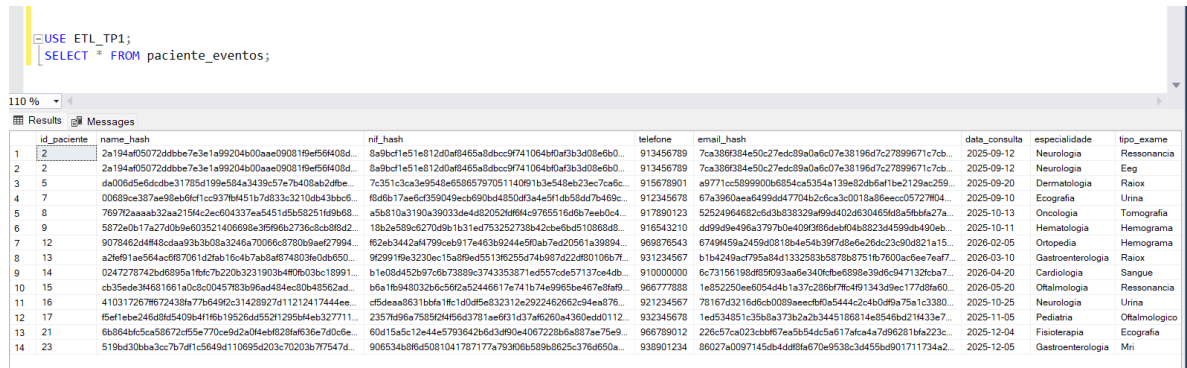
##### Descrição do fluxo

O fluxo inicia quando recebe o dataset anonimizado e consolidado que é a saída do módulo hash. Como se pode observar na *Figura 10*, o nó DB Connector (SQLServer) estabelece a ligação à base de dados. Este nó centraliza os parâmetros de conexão (endereço do servidor, credenciais e nome da base de dados), gerando uma porta de conexão que é utilizada pelos nós subsequentes. O workflow utiliza um conector específico para SQL Server.

A partir da conexão estabelecida, o fluxo ramifica-se para duas operações principais. A primeira operação é efetuada pelo DB Table Creator, que é configurado para, se necessário, criar o esquema da tabela de destino na base de dados, utilizando a estrutura de colunas do dataset de entrada. Esta etapa garante que a base de dados está preparada para receber os dados, definindo tipos de dados e restrições.

A segunda operação é realizada pelo DB Writer, o nó de escrita final. Este nó recebe tanto os dados a carregar (via a porta de dados do Component Input) como a porta de conexão (do DB Connector). O DB Writer executa a inserção dos registos no banco de dados, confirmando a conclusão do processo de Load (Armazenamento).

O Component Output finaliza o módulo, indicando a conclusão do armazenamento dos dados no sistema de armazenamento central.



USE ETL\_TP1;  
SELECT \* FROM paciente\_eventos;

id_paciente	name_hash	nif_hash	telefone	email_hash	data_consulta	especialidade	tipo_exame
1	2a194a05072dddbce7e3e1a99204b00aae09081f9e56408d...	8a9bd1e51e812a0bf0465a8dbcc9741064b0f3d3d09e6b0...	913456789	7ca3869384e50c27edc89a0a6c07e38196d7c27899671c7b...	2025-09-12	Neurologia	Ressonancia
2	2a194a05072dddbce7e3e1a99204b00aae09081f9e56408d...	8a9bd1e51e812a0bf0465a8dbcc9741064b0f3d3d09e6b0...	913456789	7ca3869384e50c27edc89a0a6c07e38196d7c27899671c7b...	2025-09-12	Neurologia	Eng
3	da006e5e6dcdbe31785d199a504a3439c57e7b408ab2d8e...	7c351c3a3e9549e6586579705114091b3e548eb23ec7ca6c...	915678901	a9771cc5899900b6854ca3535a139e82db6ef18ec2129ac259...	2025-09-20	Dermatologia	Raiox
4	00689ce387ae8eb6f6d1cc937b6451b7d833c3210db43bb6f...	f8d8b17ae5cd359049ecb690bd4850df3a4e5f1db5dd7b469c...	912345678	67a3960aae6499d447704b2c6ca3c0018a86eecc057270d4...	2025-09-10	Ecografia	Urina
5	5872e0b17a27d0b9e603521406699e3f9f96b2736c8cb89d2...	a5b810a3190a390333d4d52052d6f94c9765516d6b7eeb0c4...	917890123	52524964682e6d3838329af994d02a6304659d8a5fb6a27a...	2025-10-13	Oncologia	Tomografia
6	907846244f8cd9a3b363b08a3246a70066c8780b9aef27994...	18b2e589c6270d9b1b31ed753252738b42cbe6b510888d8...	916543210	dd99d9e496a3797b0e409f968deb04db8823d44599db490eb...	2025-10-11	Hematologia	Hemograma
7	a2ef91ae504ac687061d2fab16c4b7ab8a9748039e0db650...	f62eb3442a4799ceb917e463b9244e5f0ab7ed20561a39894...	969876543	6749459a2459d0818b4e54b39f7d8e6e26dc23c90d821a15...	2026-02-05	Ortopedia	Hemograma
8	0247278742b68995a1b6c7b220b3231903b4f0b03bc18991...	9c2991f9e3230ec15a89ed55136255d74b987d22d80106b7...	931234567	b1b4249acdf795a84d1332583b5878b6751b7600ac6ee7eaf7...	2026-03-10	Gastroenterologia	Raiox
9	cb35ede34681661a0c8c0045783b96ad484ec80b48562ad...	b1e084452b97c6b73889c3743353871ed557ede57137ce4db...	910000000	6c73156198d85093aa6e340fcbef6898e39d6c947132bca7...	2026-04-20	Cardiologia	Sangue
10	4103172678672438fa77b64992c31428927d11212417444ee...	b6a1f6948032b6c5692e52446617e741b74e9965b6e467e8fa9...	966777888	1e852250ee60544ab1a37c286bf76c491343d9ec177d8f60...	2026-05-20	Oftalmologia	Ressonancia
11	f5ef1ebe246d8d5409a4f186b19526dd5521295844eb327711...	cf5dea8631bbfa18c1d0d95e832312e292246266c94ea876...	921234567	78167d3216d6cb0089eeecbf0a5444c2c4b0d9a75a1c3380...	2025-10-25	Neurologia	Urina
12	6b864bdc8a58672c95e770ce9d2a04eb828a636e7d0c6e...	2357696a758524566d3781ae6931d37a6260a4360ed40112...	932345678	1ed534851c35b8a373b2a2b3445186814e8546bd21433e7...	2025-11-05	Pediatria	Oftalmológico
13	519bd30bba3cc7b7af1c56494110695a203c70203b77547d...	60d15a5c12e44e5793642b6d3a990e4067228b6a887ae79e9...	966789012	226c57ea023dbaf67ea5b54dc5a617afca4a7d96281bfa223c...	2025-12-04	Fisioterapia	Ecografia
14		906534869d5081041787177a79306b589b8625c376d695a...	938901234	86027a0097145b4d4d89a670e9538c3d459bd901711734a2...	2025-12-05	Gastroenterologia	Mn

Figura 11 - Tabela paciente\_examens (sql)

Na *Figura 11* podemos ver a tabela paciente\_examens no SQL Server, que representa a saída final do workflow ETL após a conclusão da fase de armazenamento. A visualização direta desta tabela, obtida através do comando `SELECT * FROM paciente_examens;`, atesta que o dataset consolidado, limpo e estruturado foi gravado corretamente no destino.

Esta imagem também serve como prova da eficácia da componente HASH. As colunas de identificação pessoal originais foram substituídas pelas suas versões anonimizadas: name\_hash, nif\_hash e email\_hash. O conteúdo destas colunas, composto por hashes (resultantes da função SHA-256), demonstra que a pseudonimização dos dados foi aplicada conforme os requisitos de proteção.

Os dados clínicos e de evento (data\_consulta, especialidade, tipo\_exame) permanecem legíveis e inalterados, validando a seletividade da transformação implementada. O resultado final apresentado na base de dados é um dataset pronto para consumo analítico e em conformidade com as políticas de segurança de dados.



## 5. Jobs – knime

A fase de Jobs (Tarefas) engloba os workflows independentes que, embora logicamente dependentes dos dados processados pelo fluxo ETL principal, são executados de forma sequencial e autónoma. O objetivo desta arquitetura é isolar funções de elevada complexidade ou de natureza transacional (como o envio de e-mails) do processamento de dados em massa, promovendo a modularidade, escalabilidade e facilidade de manutenção do sistema. O primeiro job essencial é o Workflow Email, detalhado nesta secção.

### 5.1. Arquitetura: Justificação da Comunicação

O design de um sistema implica a escolha dos mecanismos de comunicação entre módulos. Esta secção detalha a análise técnica subjacente à decisão de arquitetura adotada, focando-se na forma como o workflow principal (ETL) e o workflow de notificação (e-mail) interagem.

#### 5.1.1. Limitações na Ligação Direta (ETL -> Email)

O projeto inicial separava o fluxo principal ETL (consulta\_examenes) do fluxo de envio de notificações (email/). O objetivo era acionar o workflow de email diretamente, transferindo os dados processados.

Contudo, a integração direta entre workflows no KNIME apresentou um desafio. Ao utilizar o nó Call Workflow para tentar passar a tabela completa de dados, verificou-se que a tabela final de consulta\_examenes não conseguia passar de forma eficiente para o workflow de email/.

#### 5.1.2. Solução Ficheiros e Variáveis de Fluxo

Em vez de forçar a integração direta de workflows, optou-se por uma solução modular, utilizando mecanismos de comunicação que asseguram a independência de cada módulo.

#### Comunicação Baseada em Ficheiros

O workflow ETL principal processa e armazena os dados (incluindo os de email) em uma tabela auxiliar (data/tables/email\_table.csv). O workflow de email é, então, executado de forma sequencial e autónoma, lendo o ficheiro auxiliar para obter os destinatários e os resultados. Este método garante que cada workflow é autónomo, com uma dependência direta e auditável (o ficheiro CSV).

## 5.2. Workflow Email (Envio de Notificações)

Este workflow atua como um job de processamento dedicado, sendo o responsável por executar o envio de notificações. O seu papel é utilizar a tabela auxiliar que foi preparada e gravada pelo módulo EMAIL (descrito no ponto 4.1.4) no workflow principal, e processar cada registo individualmente.

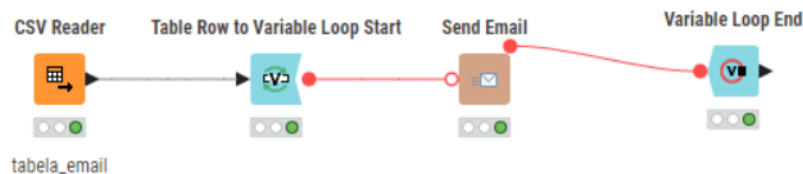


Figura 12 - Workflow Email

### Nós utilizados:

- CSV Reader (table\_email)
- Table Row to Variable Loop Start
- Send Email
- Variable Loop End

### Descrição do fluxo

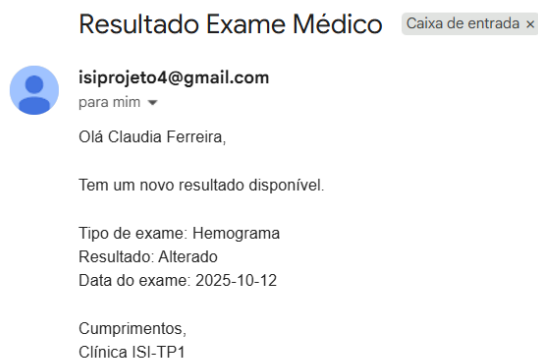
Como se pode ver na *Figura 12* fluxo começa com o nó CSV Reader. Este nó é configurado para ler a tabela auxiliar (tabela\_email), que é o ficheiro CSV contendo os registos de pacientes elegíveis para notificação (pacientes com emails de teste verdadeiros).

Em seguida, o workflow utiliza uma estrutura de repetição (loop) para processar cada linha do dataset individualmente. O nó Table Row to Variable Loop Start inicia este ciclo. A função deste nó é converter cada linha da tabela lida em variáveis de fluxo separadas (como email\_destino, nome\_paciente, etc.) que podem ser acedidas pelos nós seguintes dentro do loop.

O nó Send Email está posicionado dentro do loop e é a operação de envio. Para cada iteração, este nó utiliza as variáveis de fluxo criadas, configurando dinamicamente: o endereço de destino (o e-mail real de teste) e o corpo e o assunto da mensagem (que incluem o resultado, o tipo e a data do exame).

Finalmente, o nó Variable Loop End marca o fim da estrutura de repetição. Este nó fecha o loop assim que a última linha da tabela (tabela\_email) for processada, confirmando o envio de todas as notificações.

### 5.2.1. Validação do Envio (Mensagem Final)



*Figura 13 - Email enviado*

A *Figura 13* demonstra o resultado da execução do workflow de envio de notificações, atestando o sucesso do job e confirmando o funcionamento do nó Send Email dentro da estrutura de repetição (loop). O conteúdo da mensagem recebida valida o uso dinâmico das variáveis de fluxo, criadas pelo Table Row to Variable Loop Start.



## **6. Pentaho Kettle (Spoon)**

### **6.1. Estratégia Utilizada - Pentaho Kettle (Spoon)**

A estratégia adotada para a integração de dados clínicos baseia-se no desenvolvimento de um processo ETL (Extract, Transform, Load) automatizado, estruturado em transformações principais para o processamento de dados e em Jobs para a orquestração do fluxo de trabalho e armazenamento de dados. O processo inclui steps (operadores) e transformações específicas para garantir a qualidade, a consistência e a anonimização dos dados, seguindo um pipeline sequencial e rastreável.

#### **6.1.1. Importação e Exportação de Dados**

A extração de dados incidiu sobre duas fontes principais: as marcações de consultas médicas, em formato XML, e os resultados laboratoriais, em formato CSV. O processo iniciou-se com a leitura e parsing dos ficheiros estruturados, utilizando steps adequados para cada formato. Para os dados XML, recorreu-se ao step XML Input (Transformação Consultas), que extrai os campos relevantes das consultas, enquanto os ficheiros CSV foram processados com o step CSV Input (Transformação Exames), garantindo a leitura correta de todos os registos laboratoriais.

O processo de exportação de dados estruturados lida com a separação dos registos válidos e inválidos. Os registos que falham a validação são separados e gravados em ficheiros específicos (e.g., Consultas\_Rejeitados) através do step Text File Output, permitindo a auditoria e controlo da qualidade dos dados. Os registos válidos seguem diretamente no fluxo de dados (pipeline) para as etapas de transformação e integração subsequentes. A execução de todas as etapas é registada no ficheiro de log gerado pela ferramenta Kitchen.bat, assegurando a rastreabilidade e monitorização do processo ETL. Esta metodologia permite uma extração consistente e controlada, preparando os dados para a fase de processamento e integração.

#### **6.1.2. Transformação de Dados**

A transformação de dados consistiu na validação, normalização, integração e anonimização dos registos clínicos, preparando-os para o armazenamento. A validação utilizou expressões regulares aplicadas nos steps Filter Rows (e.g., validarConsultas e ValidarExames) para verificar o formato de campos críticos como datas, NIF e e-mails, e identificar os registos inválidos.

A integração de dados realizou-se através do step Merge join, que executa a junção de eventos clínicos por paciente, combinando as informações de consultas e exames num único conjunto de dados (Transformação join).

Foram aplicadas operações de normalização de campos textuais, conversão de formatos de data e criação de colunas derivadas para manter a uniformidade e facilitar consultas. Estas operações foram implementadas através do step Modified JavaScript Value (Passar para hash), utilizando scripting para aplicar a lógica de limpeza e formatação.

A anonimização dos dados sensíveis, nomeadamente nome, e-mail e NIF, foi feita mediante a aplicação de hashing SHA-256 no step Passar para hash (Transformação join), preservando a análise estatística sem expor a informação pessoal. Todas as operações foram implementadas com steps da plataforma Pentaho Kettle, permitindo uma execução automatizada e rastreável de todo o processo.

### **6.1.3. Armazenamento e Distribuição**

Após o processamento e a anonimização, os dados transformados foram gravados no repositório central em Microsoft SQL Server, utilizando o step Table output. Este repositório final garante a integridade e consistência, permitindo a realização de consultas uniformizadas e fornecendo suporte aos processos de análise e decisão.

O processo incluiu igualmente a integração com serviços externos para distribuição dos dados. O workflow utiliza o step JSON output e o step HTTP post (Transformação EXPORTAR\_RESULTADOS) para simular o envio dos resultados em JSON para uma API externa. Adicionalmente, o Job inclui a funcionalidade de notificação automática dos pacientes por e-mail sobre novos resultados laboratoriais, utilizando o step Mail. Esta abordagem permite disponibilizar os dados transformados para utilização interna e para comunicação com sistemas externos, mantendo a rastreabilidade e o controlo do fluxo ETL.

### **6.1.4. Ferramentas e Componentes**

O processo ETL foi implementado em Pentaho Kettle (Spoon), utilizando transformações visuais para organizar a extração, processamento e armazenamento dos dados clínicos. Foram utilizados steps essenciais, incluindo XML Input e CSV Input para a leitura de ficheiros; Merge join para a junção de registos por paciente; Modified JavaScript Value para normalização de campos e anonimização de dados sensíveis; e Table output para gravação na base de dados. A orquestração e o controlo do fluxo de trabalho foram geridos através de Jobs, garantindo que as transformações fossem executadas na ordem e sob as condições corretas. O Pentaho Kettle permite a execução automatizada e rastreável de todas as etapas do processo, assegurando a consistência e a integridade dos dados e facilitando a monitorização e auditoria.

### 6.1.5. Arquitetura do Sistema ETL no Pentaho Kettle

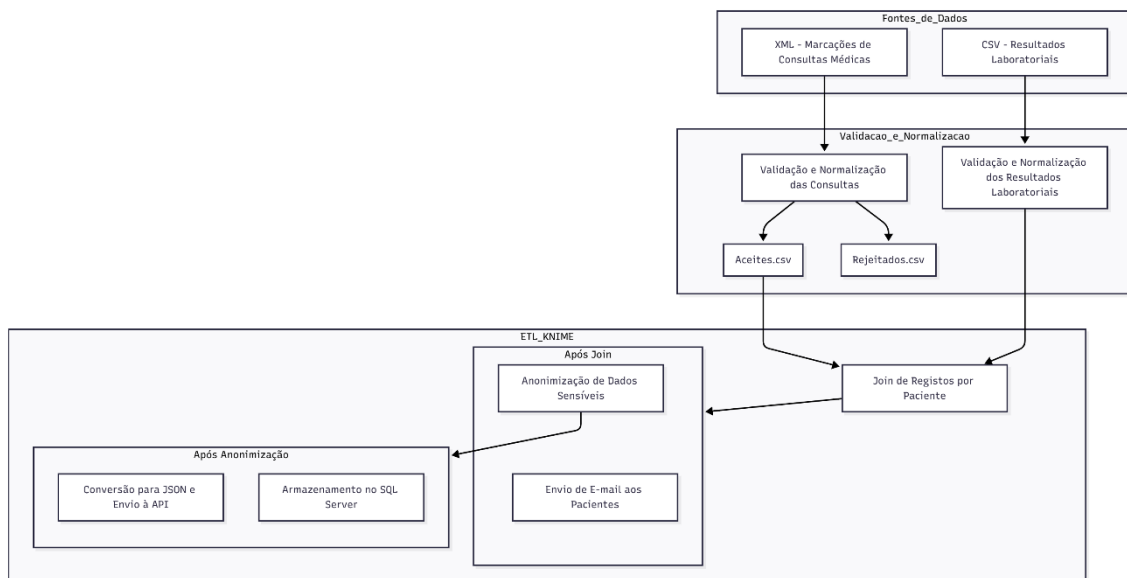


Figura 14 - Diagrama da arquitetura Pentaho kettle

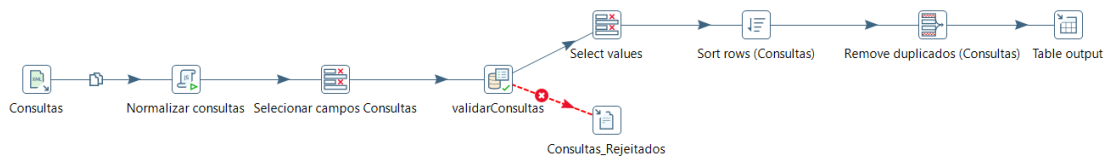
O diagrama do Job Master (ETL\_Master.kjb) representa a arquitetura do sistema de integração de dados clínicos implementado no Pentaho Kettle. As fontes de dados (XML e CSV) são verificadas nos steps Checks if files exist antes de qualquer processamento.

As transformações Consultas e Exames executam o pré-processamento, validação e limpeza dos dados. Os dados limpos são combinados através da transformação join (que inclui o Merge join e o hashing), consolidando a informação clínica.

Após a integração, o fluxo de dados segue para as operações de distribuição: o módulo email é acionado para o envio de notificações, e o módulo EXPORTAR\_RESULTADOS efetua a conversão para JSON e o envio à API externa. Paralelamente ou sequencialmente, o dataset anonimizado é gravado no Microsoft SQL Server. A arquitetura modular, baseada em Transformações e orquestrada pelo Job Master, permite a execução automatizada de todas as etapas, mantendo a consistência, rastreabilidade e integridade dos dados clínicos.

## 6.2. Transformação-Consultas

Esta transformação executa a extração (Extract) e o pré-processamento inicial dos dados de marcações de consultas médicas, cuja fonte primária é o ficheiro estruturado em XML (consultas.xml). O fluxo utiliza os seguintes nós (Steps): XML Input (Consultas) para a leitura do ficheiro XML de origem; Normalizar consultas (String Operations) para limpeza e normalização inicial; Selecionar campos Consultas (Select Values) para renomear e reordenar colunas; e validarConsultas (Filter Rows) para aplicar as regras de validação.



**Figura 15 - Transformação consultas**

O fluxo inicia com o XML Input, que carrega a estrutura de dados e extrai as colunas chave, como id\_paciente. A limpeza e a normalização de dados ocorrem no nó Normalizar consultas, utilizando scripting (JavaScript) para implementar a seguinte lógica de transformação:

1. Limpeza e Normalização de Texto: Os campos de texto simples (nome, especialidade) são convertidos para maiúsculas após a remoção de espaços em branco nas extremidades (limparTexto). O nome é submetido a um processo adicional de remoção de acentos e caracteres diacríticos (limparNome).
2. Normalização Numérica: Os campos nif e telefone são limpos de todos os caracteres não-numéricos (normalizarNif, normalizarTelefone) e o id\_paciente é explicitamente convertido para o tipo numérico (integer).
3. Normalização de Email: O campo email é convertido para minúsculas.
4. Normalização de Data: O campo data\_consulta é normalizado e formatado no padrão 'YYYY-MM-DD' para garantir a consistência de tipo e formato (normalizarData).

A validação dos dados críticos (nif, telefone e email) é aplicada através de expressões regulares no nó validarConsultas (Filter Rows). Os registos que não cumprem as regras de validação são direcionados para o passo Consultas\_Rejeitados (Text File Output), que persiste os registos inválidos para auditoria. Os registos válidos seguem para a próxima etapa. Antes de avançar, os dados passam por um nó Select Values para reorganizar e filtrar as colunas essenciais, seguido por Sort rows e Remove duplicados (Remove Duplicate Rows) para garantir a unicidade e a ordem dos registos. Os dados válidos e limpos são direcionados para o fluxo principal.



### 6.3. Transformação-Exames

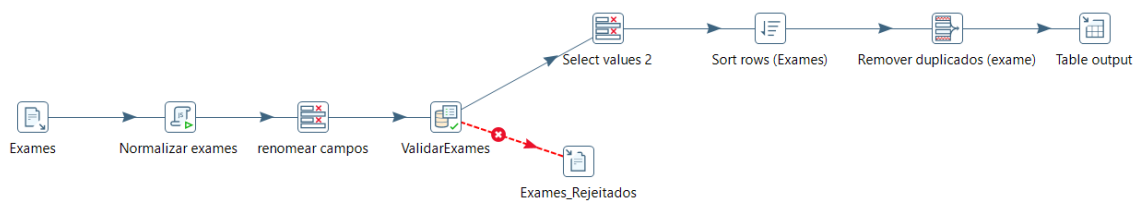


Figura 16 - Transformação Exames

Esta transformação executa a extração e a pré-transformação dos resultados laboratoriais, utilizando como fonte o ficheiro CSV (exames.csv). O objetivo é estruturar os dados e garantir a conversão correta dos tipos de dados essenciais.

O fluxo, visível na imagem acima, inicia com o nó Exames (CSV Input) para a leitura dos dados. Segue-se o nó Normalizar exames, onde a limpeza de espaços e a normalização são aplicadas a colunas chave, como id\_paciente, tipo\_exame, resultado e data\_exame. Um nó renomear campos (Select Values) é utilizado para padronizar os nomes das colunas de acordo com o esquema de destino.

A validação dos dados de exames ocorre no nó ValidarExames (Filter Rows). A validação dos dados críticos, como a data, é aplicada através de expressões regulares (Regex). Os registos que falham a validação são direcionados para o nó Exames\_Rejeitados (Text File Output), enquanto os registos válidos avançam no fluxo.

O fluxo de dados válidos passa então pelo nó Select values 2, que filtra e organiza as colunas. Em seguida, o nó Sort rows (Exames) ordena os registos e o Remover duplicados (exames) garante a unicidade. O dataset processado e com tipos de dados corrigidos é então direcionado para o Table output, aguardando a integração.

### 6.4. Transformação - Integração dos Dados (Merge Join)

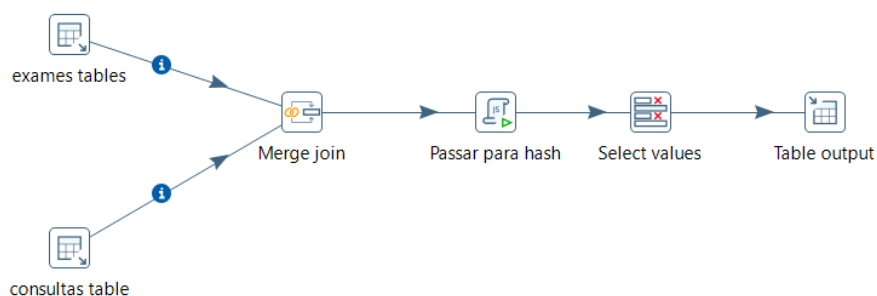


Figura 17 - Transformação- Integração de dados

Esta transformação executa a fase de integração dos dados, consolidando as informações provenientes das transformações de Consultas e Exames que já foram previamente extraídas, limpas e validadas.

O fluxo de integração utiliza os seguintes nós (Steps): exames tables e consultas table (como inputs de fluxo de dados limpos), seguido pelo nó Merge join para realizar a junção.

O fluxo inicia-se com dois datasets de entrada. Estes datasets representam os dados já processados nas etapas anteriores de Extract e Transform, estando prontos para serem unidos.

O nó Merge join é o elemento central desta etapa, sendo configurado para realizar uma junção entre os dois fluxos, utilizando o identificador único do paciente (id\_paciente) como chave de ligação. A operação é definida como uma junção interna (Inner Join), garantindo assim que apenas os registos de pacientes que possuem correspondência válida em ambas as fontes (uma consulta e um exame) são mantidos no dataset de saída.

O resultado do Merge join é um dataset unificado, contendo todos os campos das consultas e dos exames para o mesmo paciente. Este dataset consolidado segue então para a etapa de Passar para hash (JavaScript Value), onde é aplicada a anonimização dos dados pessoais, preparando-o para o carregamento final.

## 6.5. Transformação - Email

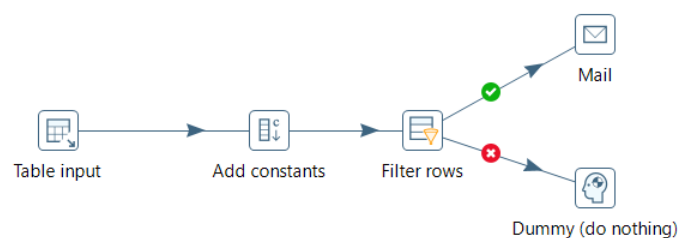


Figura 18 - Transformação Email

Esta transformação é crucial para a fase de distribuição e comunicação do processo ETL, com o objetivo de isolar, preparar e enviar notificações aos pacientes sobre os resultados dos exames recém-integrados. Esta etapa é estrategicamente executada utilizando os dados consolidados que ainda não foram submetidos à anonimização, garantindo assim o acesso aos endereços de e-mail válidos para comunicação.

O fluxo de trabalho da notificação inicia com o nó Table input. Este nó é configurado para ler a tabela auxiliar de dados (proveniente da fase de integração), que contém os registos de pacientes com a informação clínica consolidada (consulta e exame) e os

respetivos endereços de e-mail não anonimizados. Esta tabela atua como a lista de destinatários para as notificações.

A próxima etapa é o nó Add constants. Este step é essencial para introduzir dados estáticos necessários ao processo de envio de e-mail. Estes dados podem incluir as configurações do servidor SMTP (como host e porta), o endereço de e-mail do remetente (o e-mail da clínica), ou o texto padrão que será incluído no assunto ou no corpo da mensagem, isolando estas configurações da lógica de transformação de dados.

Em seguida, o fluxo de dados entra no nó Filter rows. Este step aplica uma condição lógica rigorosa para determinar quais os registos que são elegíveis para receber a notificação. Esta filtragem é vital para selecionar, por exemplo, apenas os e-mails de teste ou os pacientes com um resultado específico.

Os registos são divididos em dois fluxos com base no resultado da filtragem:

**Fluxo Positivo (Condição Cumprida):** Contém os registos elegíveis e é direcionado para o nó Mail. O nó Mail executa a operação de envio, sendo configurado para utilizar os campos dinâmicos do dataset (como o endereço de e-mail, o nome do paciente e os detalhes do exame) para personalizar a mensagem. Para cada registo processado, é enviada uma notificação individual, informando o paciente sobre a disponibilidade do seu resultado.

**Fluxo Negativo (Condição Não Cumprida):** Contém os registos que não passaram a condição de filtragem e é direcionado para um nó Dummy (do nothing). Este nó atua como um sumidouro, descartando estes dados do fluxo de notificação e concluindo o processo para os registos não elegíveis.

Este desenho modular garante que a funcionalidade de notificação é precisa e auditável, processando apenas os destinatários desejados através de uma lógica de split condicional clara. O sucesso desta transformação confirma a capacidade do sistema ETL de interagir com serviços externos de comunicação.

## 6.6. Transformação - EXPORTAR\_RESULTADOS



Figura 19 - Transformação EXPORTAR\_RESULTADOS

Esta transformação é projetada para preparar e exportar os dados processados para um sistema externo, simulando a comunicação com uma API através de uma requisição HTTP POST. Esta etapa garante que os dados consolidados e anonimizados, que são o resultado final do workflow ETL principal, podem ser consumidos por outras aplicações ou serviços.

O fluxo de trabalho inicia com o nó Table input. Este nó é responsável por extrair os dados já processados do repositório central (o Microsoft SQL Server, no nosso caso). Mais especificamente, ele recupera o dataset completo que inclui as informações de consultas e exames, já integradas e anonimizadas, conforme as etapas anteriores do ETL. A finalidade aqui é obter os dados que serão transmitidos.

O dataset recuperado segue para o nó JSON output. Este step é essencial para converter a estrutura tabular dos dados num formato de intercâmbio universalmente reconhecido para APIs: o JSON (JavaScript Object Notation). O nó é configurado para mapear as colunas do dataset de entrada para os elementos e a estrutura desejada no ficheiro JSON de saída. Esta conversão garante que os dados estão formatados corretamente para a comunicação com serviços web.

Com o dataset agora no formato JSON, o fluxo avança para o nó HTTP post. Este step é o componente-chave para a simulação da comunicação com a API externa. O nó HTTP post é configurado para:

- Especificar o URL do endpoint da API (o destino para onde os dados serão enviados).
- Definir o método de requisição como POST, que é adequado para o envio de dados.
- Incluir o conteúdo JSON gerado na etapa anterior no corpo da requisição.
- Tratar quaisquer cabeçalhos HTTP adicionais necessários para a autenticação ou para especificar o tipo de conteúdo (e.g., Content-Type: application/json).

Esta etapa simula efetivamente uma transação de dados real, permitindo testar a integração com o sistema downstream.

Finalmente, o nó Mostrar resultados (Select values para visualização ou um Text file output para registo) é utilizado para exibir ou armazenar a resposta da requisição HTTP POST. Isto permite validar se a comunicação com a API foi bem-sucedida, se os dados foram aceites corretamente e se houve alguma mensagem de erro ou de confirmação por parte do serviço externo. É uma etapa importante para monitorizar e depurar o processo de exportação.

## 6.7. JOB-ETL\_MASTER

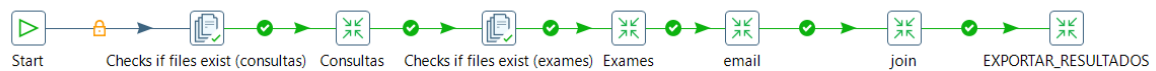


Figura 20 - JOB ETL\_MASTER

O Job principal, atua como o orquestrador do sistema de integração de dados, sendo responsável por garantir a execução sequencial e lógica de todo o processo ETL, desde a validação das fontes até a distribuição final dos resultados. O fluxo de controlo começa com o nó Start, que desencadeia a execução.

Inicialmente, o Job assegura a disponibilidade dos dados de entrada através de dois passos de verificação de pré-requisitos: o nó Checks if files exist (consultas) e o nó Checks if files exist (exames). Esta verificação é crucial para prevenir falhas e confirmar a presença dos ficheiros XML e CSV de origem antes de se iniciar qualquer processamento.

Após a confirmação da presença das fontes, o Job prossegue para a execução das transformações de pré-processamento: a transformação Consultas é acionada, processando o ficheiro XML, normalizando e validando os registos. Em seguida, a transformação Exames é executada, processando, limpando e validando os resultados laboratoriais. Estas duas etapas preparam os dados para a fase subsequente de integração.

Na fase de integração e distribuição, o *Job* executa os módulos sequencialmente:

- A transformação join é acionada, sendo responsável por consolidar os dados limpos das consultas e exames através do *Merge join* e aplicar a anonimização dos dados pessoais (*hash*).
- O módulo email (ou um Job dependente) é executado para iniciar a preparação da tabela auxiliar de notificações, isolando os registos elegíveis para envio de e-mail.
- O último passo é a execução da transformação EXPORTAR\_RESULTADOS, que conclui o processo de distribuição. Esta etapa final assegura a conversão do dataset para JSON e a simulação do envio para o serviço externo via HTTP post.

A conclusão bem-sucedida de toda esta sequência valida a funcionalidade completa e orquestrada do pipeline de integração de dados.

## 6.8. Geração e Análise dos Logs de Execução

A geração dos logs de execução é uma etapa fundamental para a auditoria, monitorização e validação do processo ETL, permitindo rastrear o estado, o desempenho e a eventual ocorrência de erros em cada etapa do Job principal.

Para executar o Job Master (ETL\_Master.kjb) fora do ambiente gráfico do Pentaho Spoon e gerar um registo detalhado da operação, foram utilizados comandos específicos no Prompt de Comando, recorrendo à ferramenta de linha de comandos Kitchen.bat:

1. Navegação para o Diretório de Execução: O comando inicial move o foco para o diretório onde o executável do Kettle (Kitchen.bat) se encontra:

```
-cd C:\Users\João\Downloads\pdi-ce-10.2.0.0-222\data-integration
```

2. Execução do Job Master com Geração de Log: O comando seguinte aciona a execução do Job principal, definindo o nível de detalhe do log e especificando o caminho e o nome do ficheiro de registo:

```
-Kitchen.bat /file:"C:\TP01_27964\dataint\ETL_Master.kjb" /level:Basic  
/logfile:"C:\TP01_27964\data\output\logs\nolog"
```

A utilização destes comandos garante que o Job (ETL\_Master.kjb) seja executado com sucesso, seguindo a sequência de transformações definida. O parâmetro /level:Basic configura o registo para capturar informações essenciais sobre o início, a conclusão e o estado de cada step do Job e das transformações associadas. O output da execução é redirecionado para o ficheiro noalog, permitindo a análise posterior e a validação do fluxo de dados através do rastreio de cada operação (verificação de ficheiros, extração, join, hash e post à API).

## 7. Dashboard

Esta secção demonstra o resultado final da arquitetura de dados através da visualização, apresentando os Dashboards criados. Estas interfaces de utilizador são alimentadas diretamente pelos dados alocados no SQL Server e servem para transformar o volume de dados brutos em informação acionável (actionable insights), permitindo a monitorização de indicadores-chave de desempenho (KPIs) e a tomada de decisões clínicas ou operacionais.

### 7.1. Resultados de Exames

Resultados de Exames

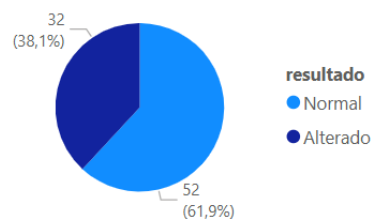


Figura 21 - Resultado de Exames

O primeiro dashboard de monitorização clínica foca-se na distribuição geral dos resultados dos exames. O gráfico da *Figura 21* apresenta a proporção de exames com resultado Normal e Alterado no dataset processado.

### 7.2. Resultados por Tipo de Exame

Resultados por Tipo de Exame

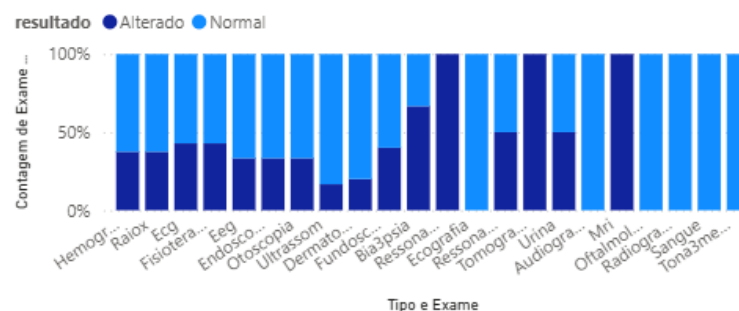


Figura 22 - Resultados por Tipo de Exame

A *Figura 22* aprofunda a análise dos resultados, apresentando a distribuição dos resultados Alterado e Normal em função de cada Tipo de Exame.

Este gráfico de barras empilhadas permite identificar os procedimentos que registam a maior percentagem de resultados anormais. É possível observar a heterogeneidade da proporção de resultados alterados (em azul-escuro) em cada categoria.

### 7.3. Consultas por Especialidade

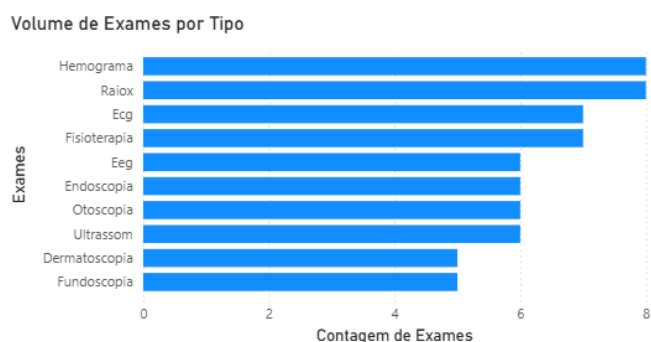


*Figura 23 - Consultas por Especialidade*

A *Figura 23* aprofunda a análise dos resultados, apresentando a distribuição global das consultas em função de cada especialidade.

Este gráfico de barras permite identificar os serviços que registam o maior volume de atendimentos no dataset. É possível observar a representatividade das diferentes áreas, como Gastroenterologia e Neurologia que lideram o número de consultas.

### 7.4. Volume de Exames por Tipo



*Figura 24 - Volume de Exames por Tipo*



A *Figura 24* quantifica o Volume de Exames por Tipo no centro de diagnóstico, apresentando uma análise da carga de trabalho.

Este gráfico de barras horizontais é fundamental para a gestão operacional e o planeamento de recursos. Permite a identificação imediata dos procedimentos que têm a maior frequência no dataset, como é o caso do Hemograma e do Raio-X.

## 8. Vídeos

O material audiovisual seguinte constitui a prova de conceito do trabalho, demonstrando a execução dos workflows desenvolvidos nas plataformas KNIME e Pentaho (spoon)

Estes vídeos validam a funcionalidade dos processos ETL em tempo real, comprovando a eficácia da arquitetura implementada. O acesso às gravações é disponibilizado através dos QR Codes apresentados abaixo.

Pentaho (spoon)	KNIME
	

## 9. Conclusão

O desenvolvimento deste trabalho permitiu compreender de forma prática e aprofundada os princípios e desafios inerentes à integração de sistemas de informação, particularmente no contexto da integração de dados clínicos. Ao longo do projeto, foi possível implementar um processo completo de ETL (Extract, Transform, Load), capaz de extrair, transformar, anonimizar e armazenar dados provenientes de diferentes fontes, assegurando a consistência, a integridade e a confidencialidade da informação tratada.

A realização do projeto nas plataformas KNIME e Pentaho Kettle (Spoon) revelou-se particularmente enriquecedora, uma vez que possibilitou não apenas a execução de dois processos de integração distintos, mas também a comparação direta entre as suas abordagens, estruturas e mecanismos de automação. Ambas as ferramentas demonstraram robustez e flexibilidade na gestão de fluxos de dados, embora com diferentes paradigmas de interface e configuração. Esta experiência permitiu adquirir uma visão crítica sobre as vantagens e limitações de cada solução, nomeadamente no que diz respeito à facilidade de monitorização, modularidade dos workflows e capacidade de integração com sistemas externos.

No que concerne ao objetivo central do trabalho — a consolidação de dados clínicos provenientes de fontes heterogêneas (ficheiros XML e CSV) —, os resultados obtidos foram plenamente satisfatórios. O sistema ETL desenvolvido foi capaz de realizar com sucesso a leitura, validação e normalização dos registos, garantindo que apenas os dados válidos e coerentes fossem integrados no repositório final em Microsoft SQL Server. A aplicação de expressões regulares, mecanismos de filtragem e validações automáticas assegurou a qualidade da informação, enquanto a utilização de algoritmos de hashing (SHA-256) garantiu a anonimização eficaz dos dados sensíveis, em conformidade com os princípios de proteção de dados pessoais e boas práticas de segurança da informação.

Para além das operações internas de processamento, foi implementada uma camada de comunicação externa, simulando a interação com uma API através de requisições HTTP POST em formato JSON e o envio automático de e-mails para notificação de resultados. Estas funcionalidades, embora simuladas, demonstram a importância da interoperabilidade entre sistemas e refletem um cenário realista de integração num ecossistema clínico digital, onde múltiplas aplicações partilham informação em tempo real para apoiar a decisão médica e a gestão operacional.

Outro aspeto relevante foi a modularização do sistema, alcançada através da separação de componentes e jobs independentes. Esta arquitetura promoveu a escalabilidade e manutenção do processo ETL, permitindo isolar fases críticas — como o envio de notificações ou a exportação de dados — sem comprometer a estabilidade do pipeline principal. Esta prática reflete um princípio essencial na engenharia de software e de dados: a separação de responsabilidades, que melhora a legibilidade, a reusabilidade e a fiabilidade das soluções implementadas.

Em termos de aprendizagem, o trabalho proporcionou uma sólida consolidação de competências técnicas nas áreas de integração de dados, modelação de fluxos ETL, automação de processos e comunicação entre sistemas distribuídos. Mais do que uma

simples implementação, o projeto exigiu análise crítica, resolução de problemas, capacidade de abstração e planejamento colaborativo entre os membros da equipa. O estudo e aplicação prática das ferramentas KNIME e Pentaho Kettle contribuíram também para reforçar a literacia tecnológica na área de Business Intelligence e Data Warehousing, competências essenciais no mercado atual.

Em suma, o projeto atingiu plenamente os objetivos propostos, culminando na criação de um pipeline ETL funcional capaz de integrar dados clínicos de forma eficiente e padronizada. O trabalho evidenciou a importância da integração de sistemas como elemento central para o sucesso das organizações modernas, especialmente no domínio da saúde, onde a qualidade e a fiabilidade dos dados podem ter impacto direto na tomada de decisão e no bem-estar dos pacientes. A experiência adquirida constitui, portanto, uma base sólida para o desenvolvimento de futuros projetos de integração e análise de dados, reforçando a ligação entre a teoria e a prática no contexto académico e profissional.

## **10. Bibliografia**

KNIME. Disponível em: <https://www.knime.com/>

Pentaho Wiki. Spoon User Guide. Disponível em: <https://pentaho-public.atlassian.net/wiki/spaces/EAI/pages/370966839/Spoon+User+Guide>

ETL - 03 - Kettle Tutorial.

ETL - 05 - Knime introduction.