Relatório Integração de Sistemas de Informação

No.12747 – João Carvalho

Licenciatura em Engenharia Sistemas Informáticos

3ºano

Barcelos | Outubro, 2024

Índice

Índice de Figuras

1. Introdução

Este projeto integra-se no âmbito da unidade curricular de Integração de Sistemas de Informação (ISI), lecionada no primeiro semestre do terceiro ano do curso de Licenciatura em Engenharia de Sistemas Informáticos. O trabalho proposto visa aprofundar o conhecimento e a aplicação de técnicas e ferramentas ETL (*Extract, Transformation, and Load*) para integração de sistemas de informação, com foco na manipulação e consolidação de dados provenientes de múltiplas fontes.

Para a realização do projeto, será utilizado o Pentaho Data Integration (Kettle), uma das principais ferramentas ETL *open-source*, que permite a transformação e o fluxo automatizado de dados entre diferentes sistemas. Esta escolha proporciona um ambiente de desenvolvimento visual, facilitando a criação de fluxos de trabalho complexos para transformar, normalizar e carregar dados, o que é fundamental para a obtenção de dados integrados e coerentes num sistema central.

Ao longo do projeto, serão abordados desafios comuns em cenários de integração, como a necessidade de extração de dados de diferentes fontes, transformação conforme regras de negócio específicas e carregamento eficiente em sistemas de destino. Este processo de experimentação e aplicação prática permitirá consolidar o conhecimento teórico em torno dos conceitos de ETL, explorando a interligação de sistemas numa infraestrutura centralizada de dados e simulando condições reais do ambiente profissional de engenharia de sistemas informáticos.

2. Desafio

Neste trabalho, a API *Humanitarian Data Exchange* (HAPI) foi utilizada como uma fonte externa de dados para integrar e enriquecer as informações no processo de ETL realizado no Pentaho Data Integration (Kettle). Através da API de refugiados do *Humanitarian Data Exchange*, obtivemos dados atualizados e detalhados sobre o número de refugiados que entraram em Portugal, juntamente com a distribuição das nacionalidades desses indivíduos.

O fluxo de trabalho foi construído de forma a fazer chamadas à API, utilizando funcionalidades para obter e carregar dados de entrada de refugiados em Portugal, categorizados por nacionalidade. Esta integração permitiu que os dados fossem transformados e preparados no Kettle para gerar métricas.

Além disso, a API trouxe uma camada adicional de automação e consistência ao processo, uma vez que possibilitou a atualização periódica dos dados sem a necessidade de importações manuais. Este método, combinado com as capacidades de transformação e manipulação de dados do Kettle, forneceu uma visão analítica importante para a análise das tendências migratórias de refugiados para Portugal, garantindo que o projeto reflita dados precisos e em tempo real.

Esta integração, portanto, proporcionou uma aplicação prática na recolha e análise de dados de sistemas externos, permitindo um maior entendimento da utilização de APIs no contexto da integração de sistemas de informação e enriquecendo o projeto com dados reais e dinâmicos.

2.1. Processos utilizados

Para a realização deste trabalho foram utilizados os seguintes processos:

- **Data Grid**
  - Permite criar e inserir dados manualmente numa transformação, sendo útil para testes, tabelas de referência e dados temporários.
- **Rest Client**
  - Permite fazer pedidos a APIs REST externas para obter, enviar ou atualizar dados, integrando essas informações ao fluxo de transformação para serem processadas ou armazenadas.
- **JSON Input**
  - Usado para ler e processar dados em formato JSON, extraindo informações específicas para serem manipuladas dentro do fluxo de transformação, como se fosse uma fonte de dados.
- **Select Values**
  - Permite selecionar, renomear, reordenar, remover e alterar o tipo de dados das colunas numa transformação, facilitando a gestão e a organização dos dados antes de etapas posteriores no fluxo.
- **Modified JavaScript Value**
  - Permite manipular e transformar dados através de código JavaScript personalizado, facilitando operações complexas e condicionais durante o processo de ETL.
- **Sort Rows**
  - Utilizado para ordenar dados numa tabela ou fluxo, facilitando o pré-processamento, a análise e a preparação para operações subsequentes.
- **Filter Rows**
  - Usado para selecionar ou descartar linhas de dados com base em condições específicas, permitindo refinar conjuntos de dados antes de outras operações no processo de ETL.

- **Microsoft Excel Writter**
    - Utilizado para exportar dados processados para arquivos no formato Excel (.xls ou .xlsx), permitindo a criação de relatórios e a exportação de dados de forma estruturada e fácil de manipular.
- **XML Output**
    - Utilizado para exportar dados processados para arquivos no formato XML, permitindo a criação de documentos estruturados que podem ser utilizados para integração com outros sistemas ou armazenamento de dados.
- **File Exists**
    - Utilizado para verificar a existência de um ficheiro ou diretório específico no sistema de ficheiros, permitindo controlar o fluxo de execução do processo de ETL com base na presença ou ausência desse ficheiro.
- **Mail**
    - Utilizado para enviar e-mails a partir do fluxo de ETL, permitindo a notificação de eventos, relatórios ou alertas com dados processados diretamente do Kettle.

3. Desenvolvimento

3.1. Transformação

Para este trabalho foi utilizado uma transformação de uma API de dados externa. O primeiro passo foi a criação de uma Data Grid de apenas uma linha com os diferentes parâmetros para executar os pedidos à API.

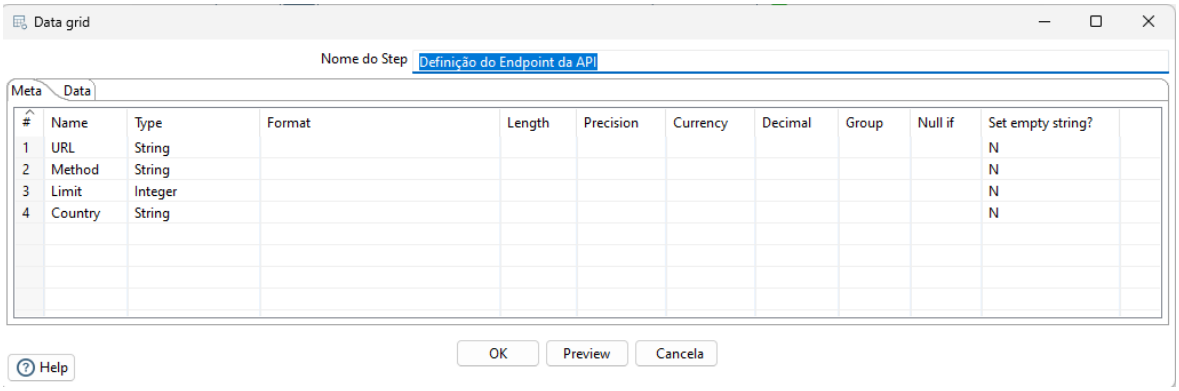

Figura 1 - Definição das colunas de dados para os pedidos à API REST

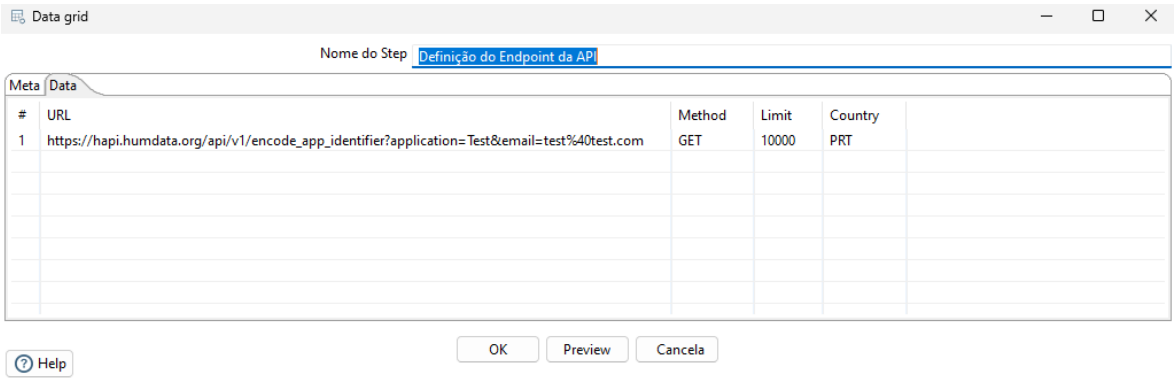

Figura 2 - Definição dos parâmetros

De seguida, foi utilizado o processo Rest Client para fazer o pedido à API para obter o identificador a usar nos diferentes endpoints, no *endpoint* definido nos parâmetros na Figura 2.

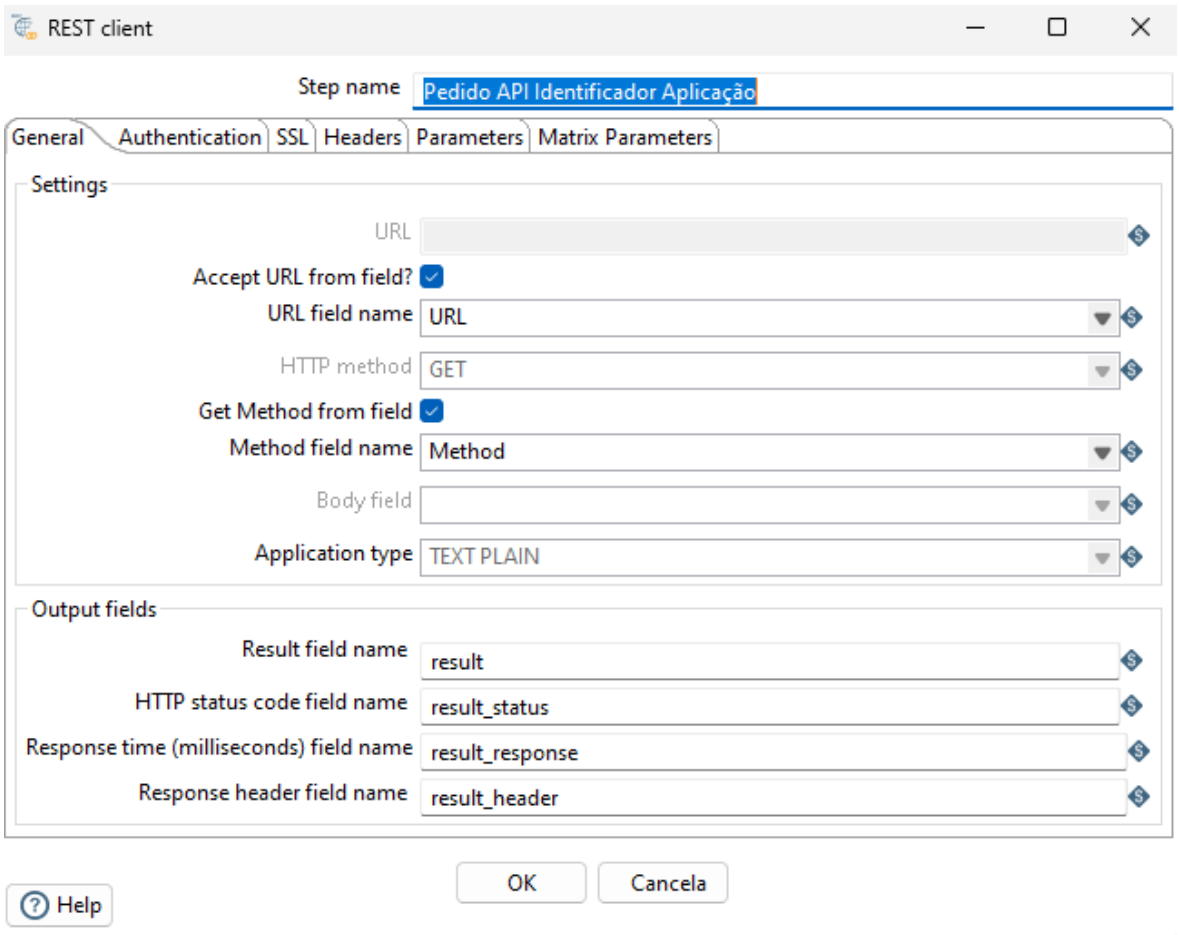

Figura 3 - Pedido à API com o URL e Method definido na Data Grid

O *URL* e o método do pedido à API, bem como os consequentes parâmetros usados nos pedidos à API tiveram de ser inseridos na Data Grid devido a uma limitação de *software* do Kettle que não permite passar parâmetros na aba 'Parameters' se o tipo de pedido for *GET*.

Após o pedido, foi utilizado o JSON Input para filtrar a resposta devolvida pela API ao pedido, utilizando para isso a variável *result* definida no passo anterior como nome de variável para o resultado, sendo depois esse valor da resposta atribuída à variável *API_KEY*.

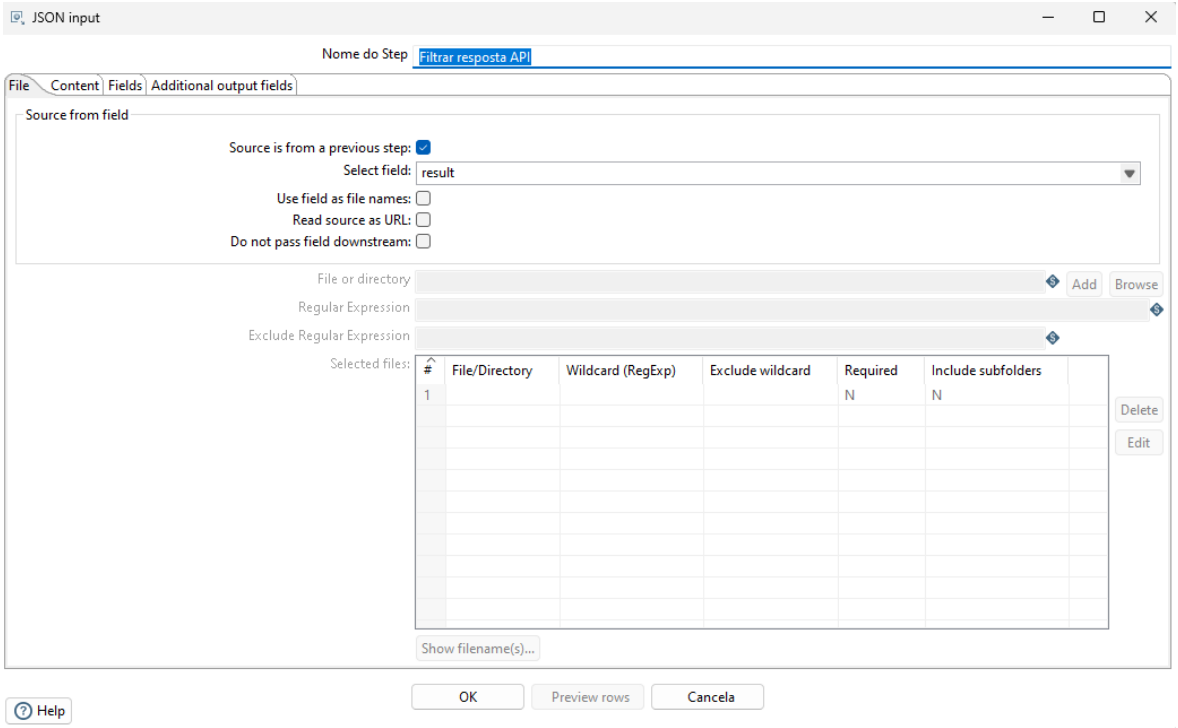

Figura 4 - Filtragem da resposta do pedido de identificador da API

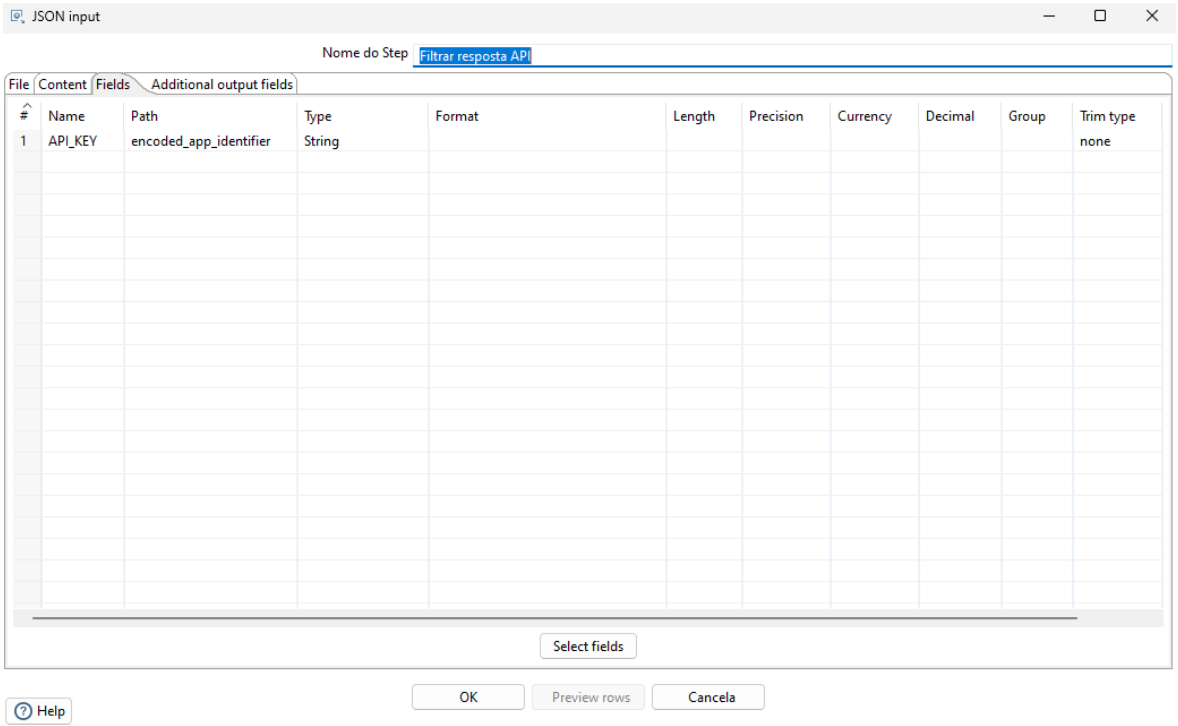

Figura 5 - Atribuição do valor da tag à variável API_KEY

Depois de obtida a resposta, é utilizado o Select Values para obter as variáveis a usar nos passos seguintes.

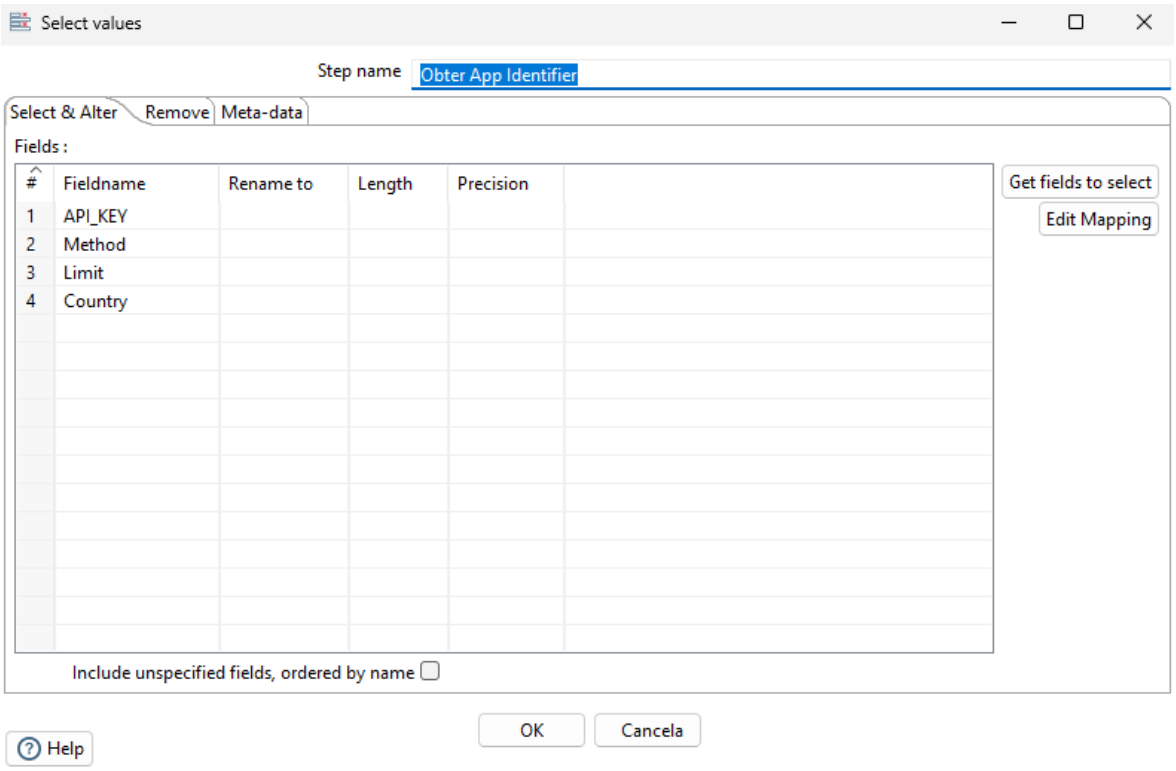

Figura 6 - Seleção de valores a utilizar nos passos seguintes

Depois de definidas as variáveis a utilizar nos próximos passos, passou-se à obtenção dos dados da API, já com a API_KEY obtida junto com os parâmetros que o *endpoint* permite, definidos pela documentação da API.

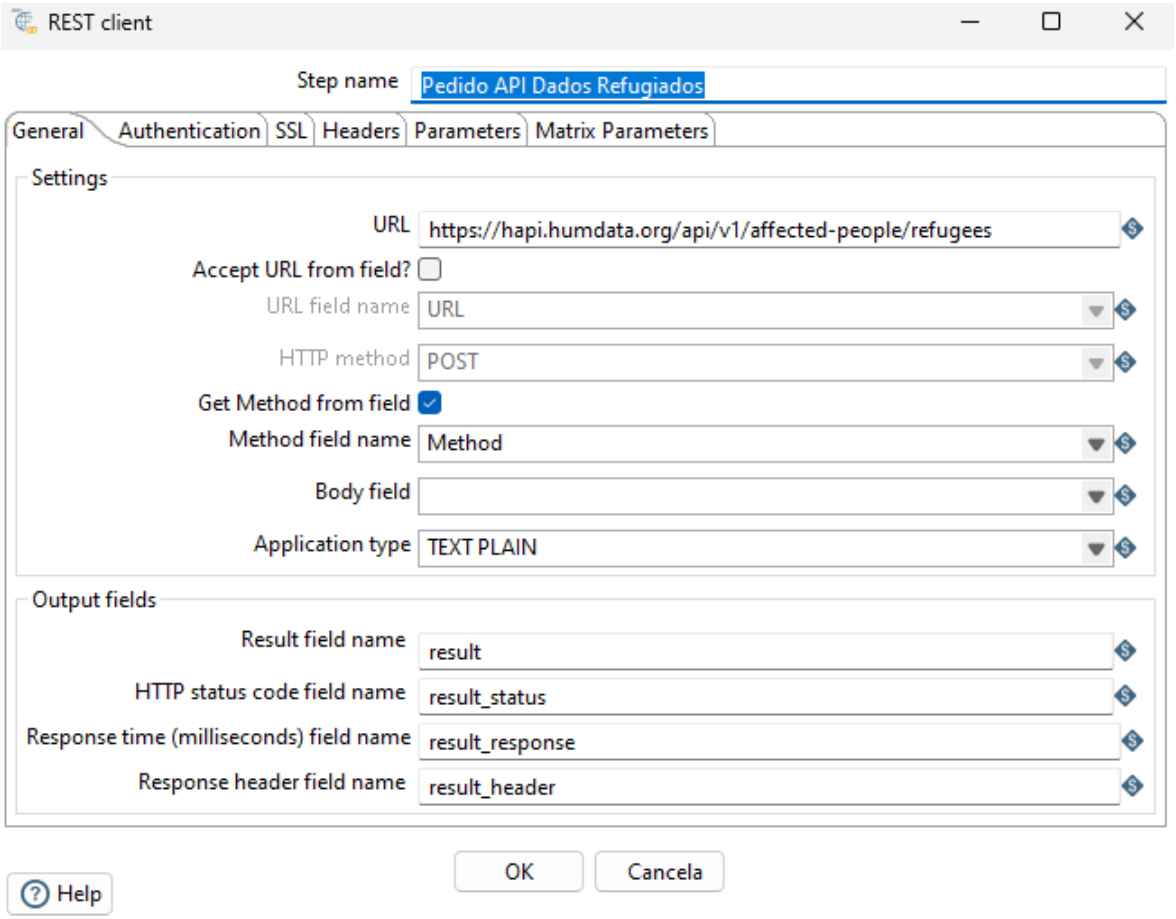

Figura 7 - Pedido à API para obtenção de dados de refugiados

Mais uma vez, foi necessário definir o método do pedido através da variável Method, para permitir que a aba 'Parameters' estivesse ativa para passar os restantes parâmetros no pedido à API.

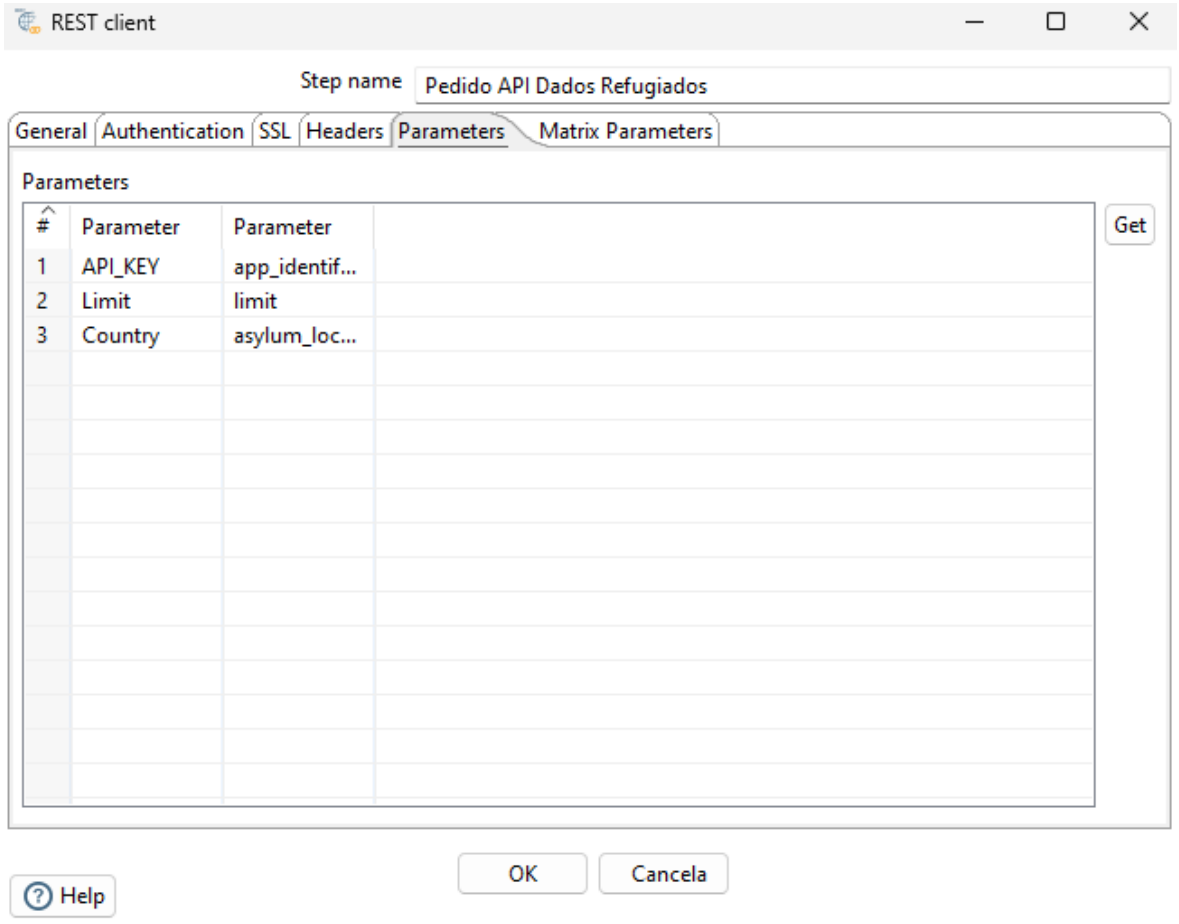

Figura 8 - Parâmetros passados no pedido à API

Tal como no passo seguinte do pedido para obter a API_KEY, utilizei novamente o JSON Input para filtrar o resultado do pedido à API, indicando novas colunas e quais os paths para a obtenção dos dados no JSON da resposta da API, novamente obtendo os dados da variável result, definida no pedido à API.

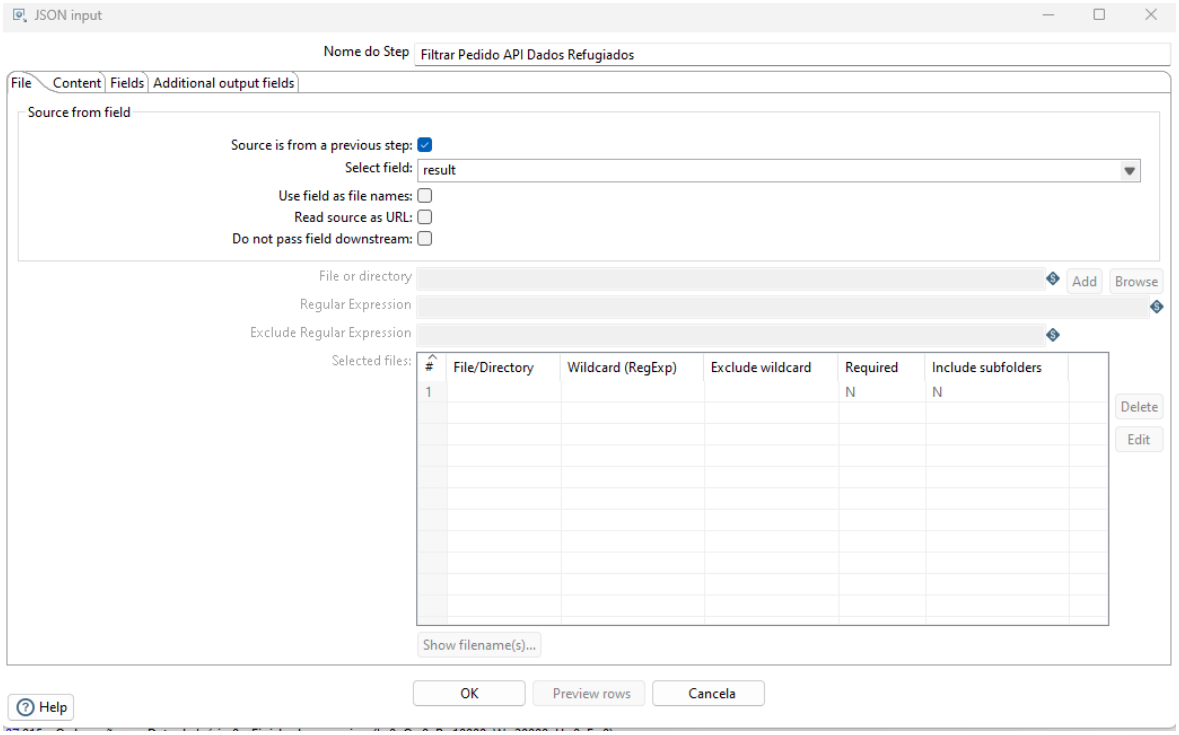

Figura 9 - Filtragem dos dados da variável result

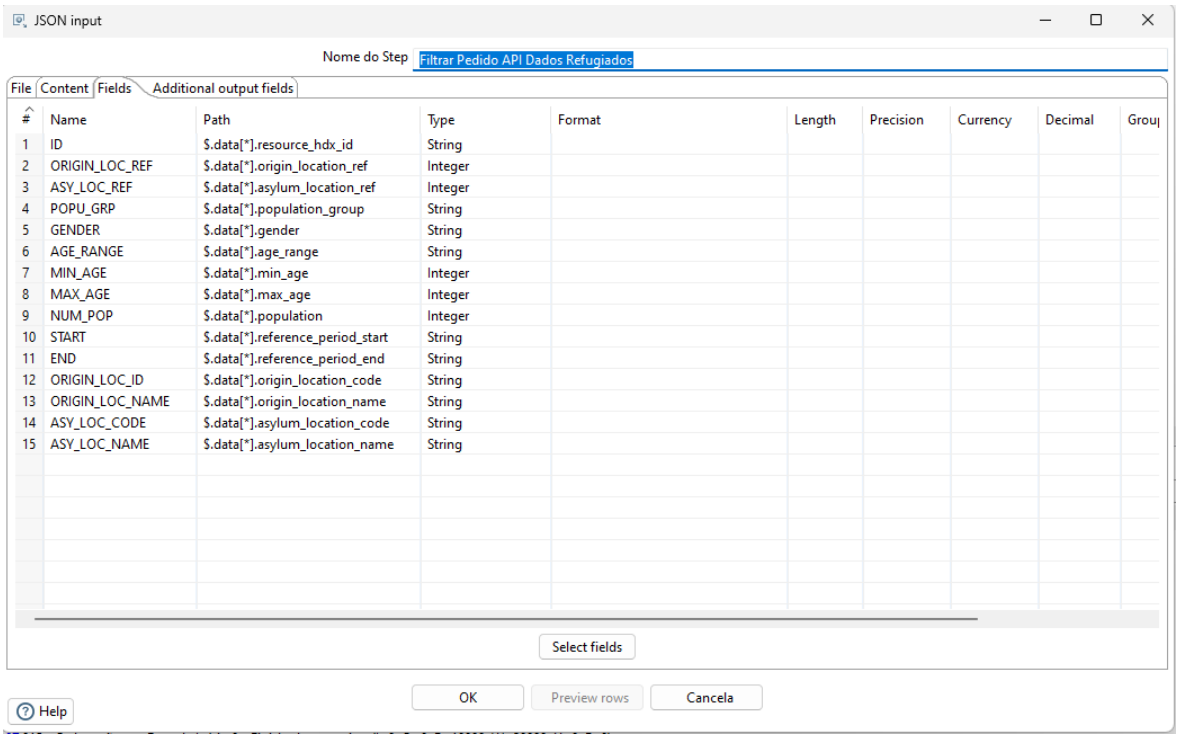

Figura 10 - Mapeamento dos dados para as variáveis correspondentes

No passo seguinte utilizei novamente o Select Values para obter as variáveis dos dados resultantes do JSON.

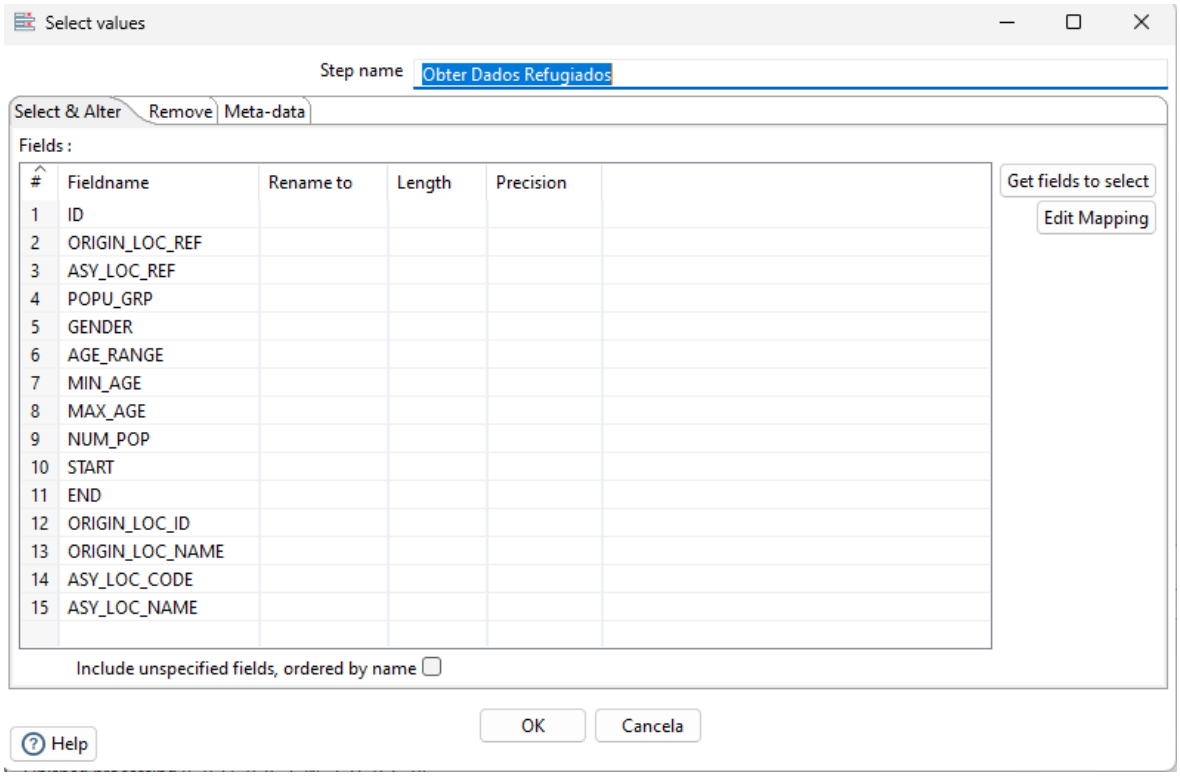

Figura 11 - Seleção dos valores JSON da resposta

Depois de obtidos os dados, procedi à normalização dos campos das datas, utilizando para isso o processo Modified Javascript Value, para substituir parte da *string* das datas.

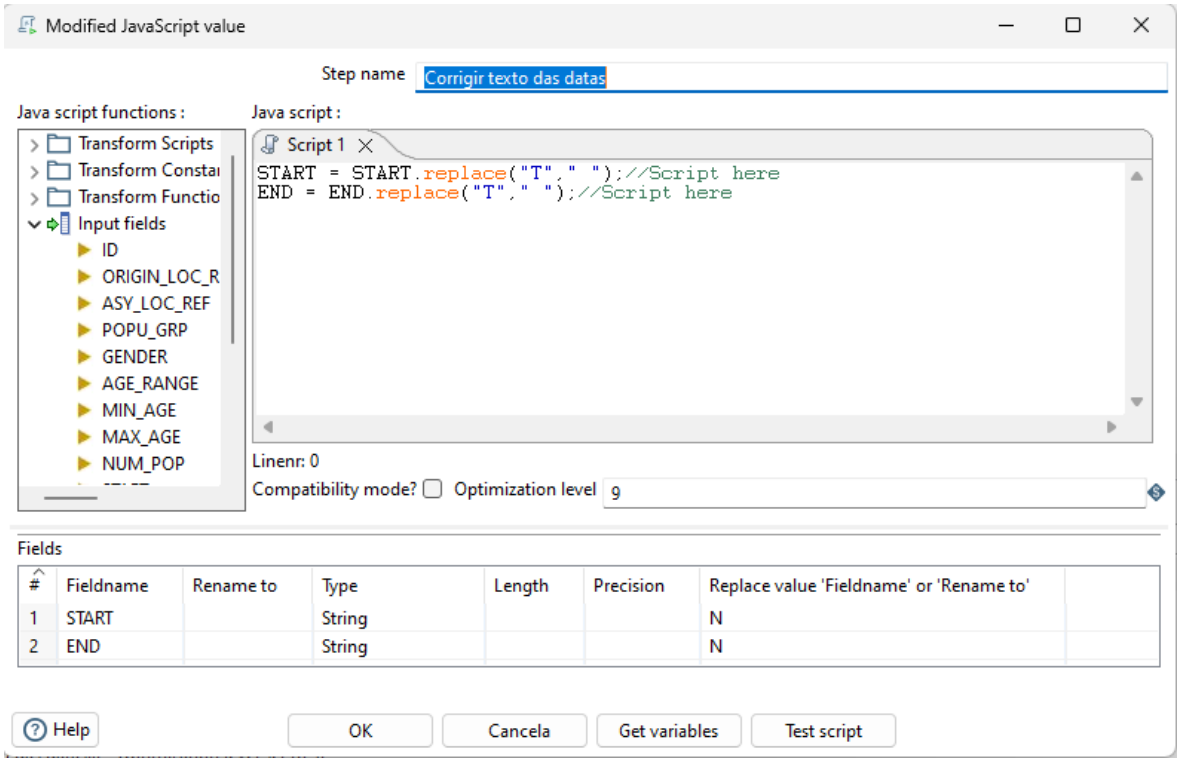

Figura 12 - Normalização das datas

Utilizei novamente o Select Value após este passo, para transformar os campos de datas para o tipo DATE na aba 'Meta-Data' com o respetivo formato.

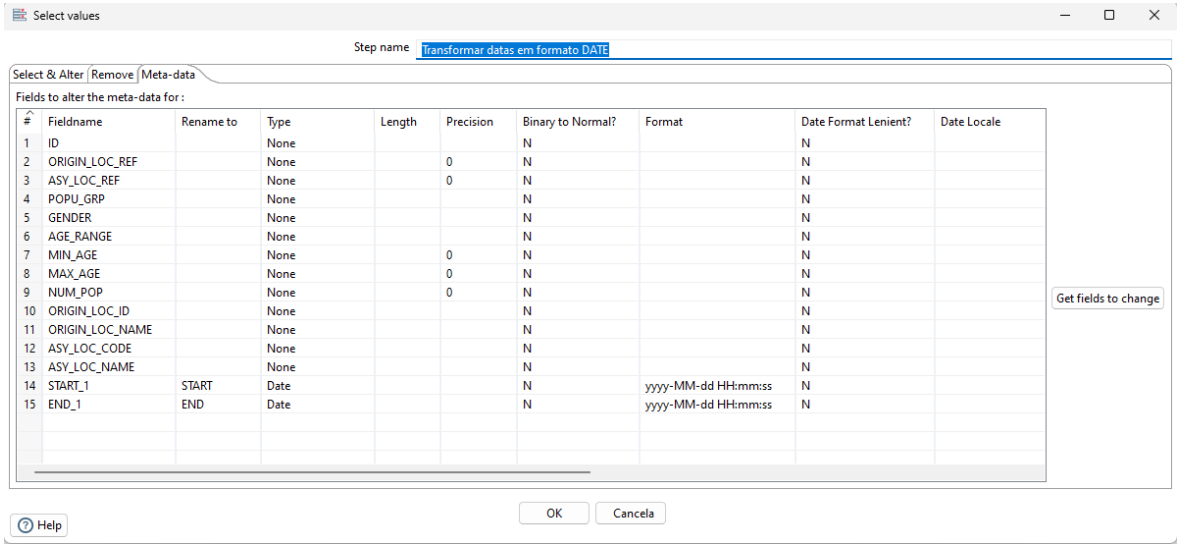

Figura 13 - Transformação dos campos das datas

Após a normalização dos campos das datas, procedi à ordenação dos dados pela coluna START.

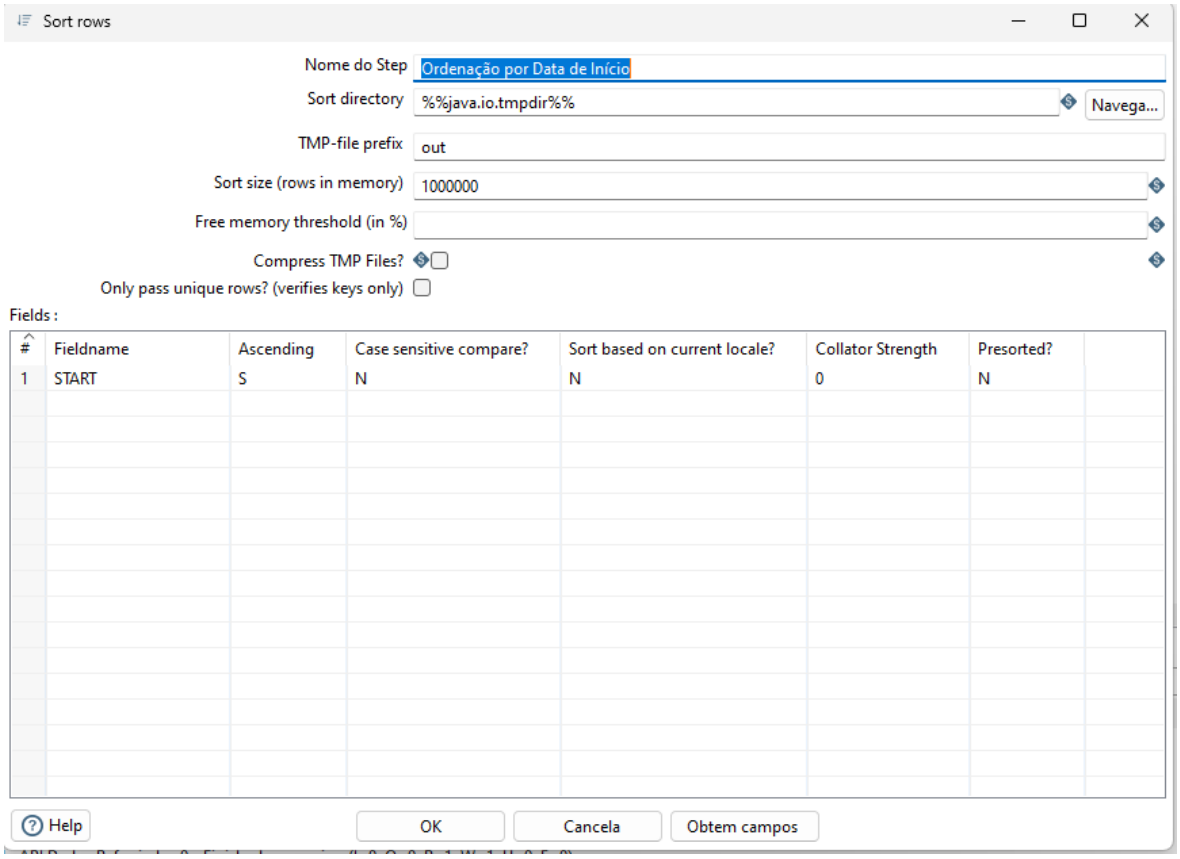

Figura 14 - Ordenação dos dados pela coluna Start

Após este passo, é gerado um ficheiro XML com todos os dados presentes e ordenados, passível de ser usado por um outro sistema caso seja necessário.

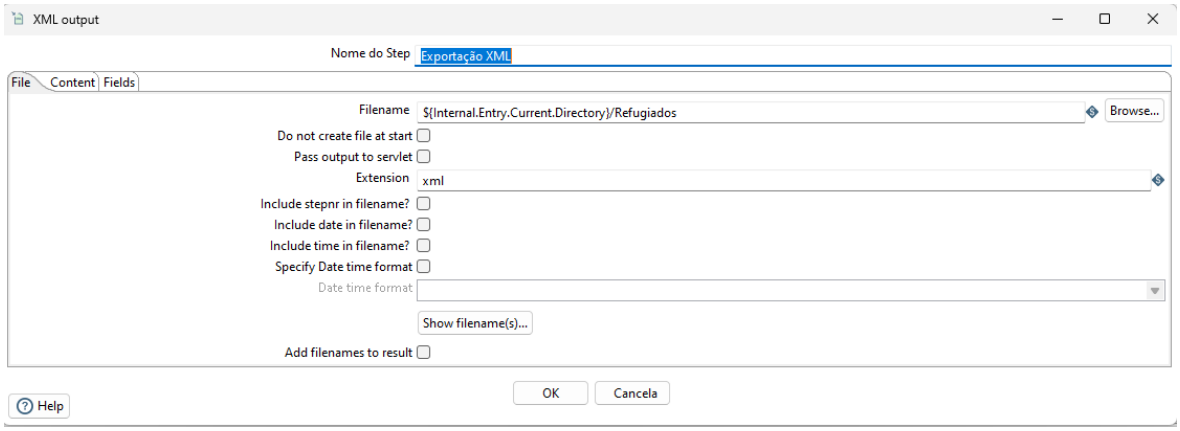

Figura 15 - Exportação XML dos dados

Apliquei de seguida uma filtragem aos dados presentes para demonstrar que é possível aprimorar os dados para obtermos apenas os dados necessários nas condições que desejámos.

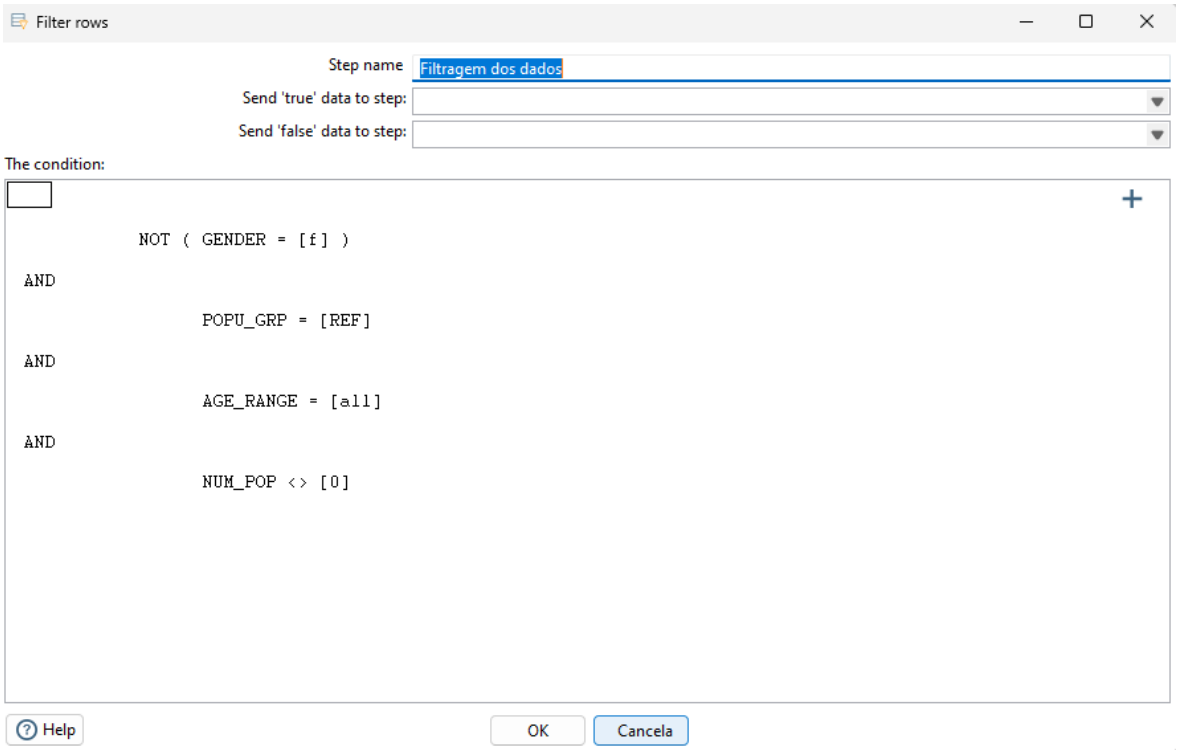

Figura 16 - Filtragem dos dados

Após a filtragem, seguiu-se a exportação dos dados para um ficheiro de formato Excel, para uma melhor compreensão e gestão por parte dos utilizadores que desejem ter estes dados.

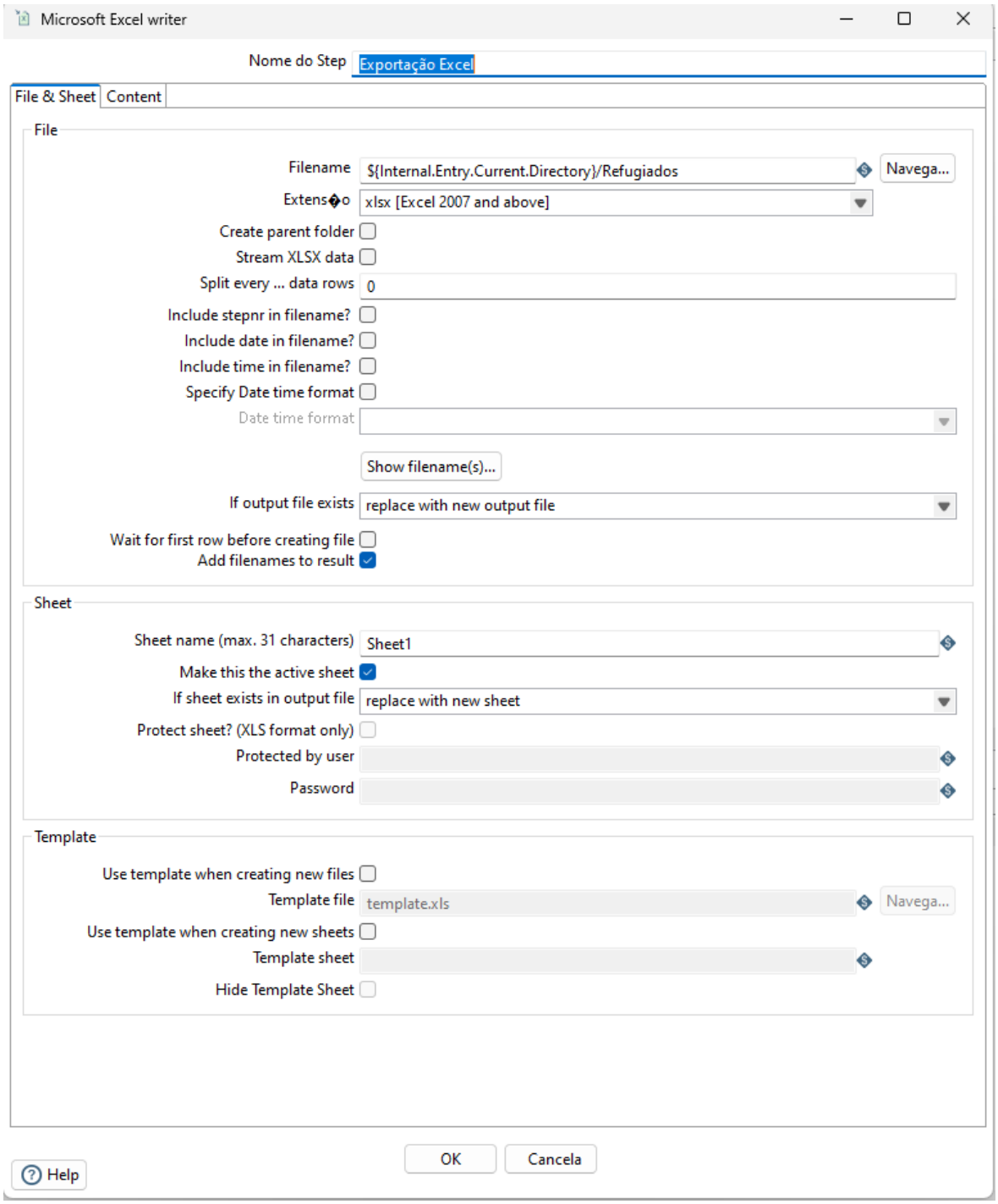

Figura 17 - Exportação Excel

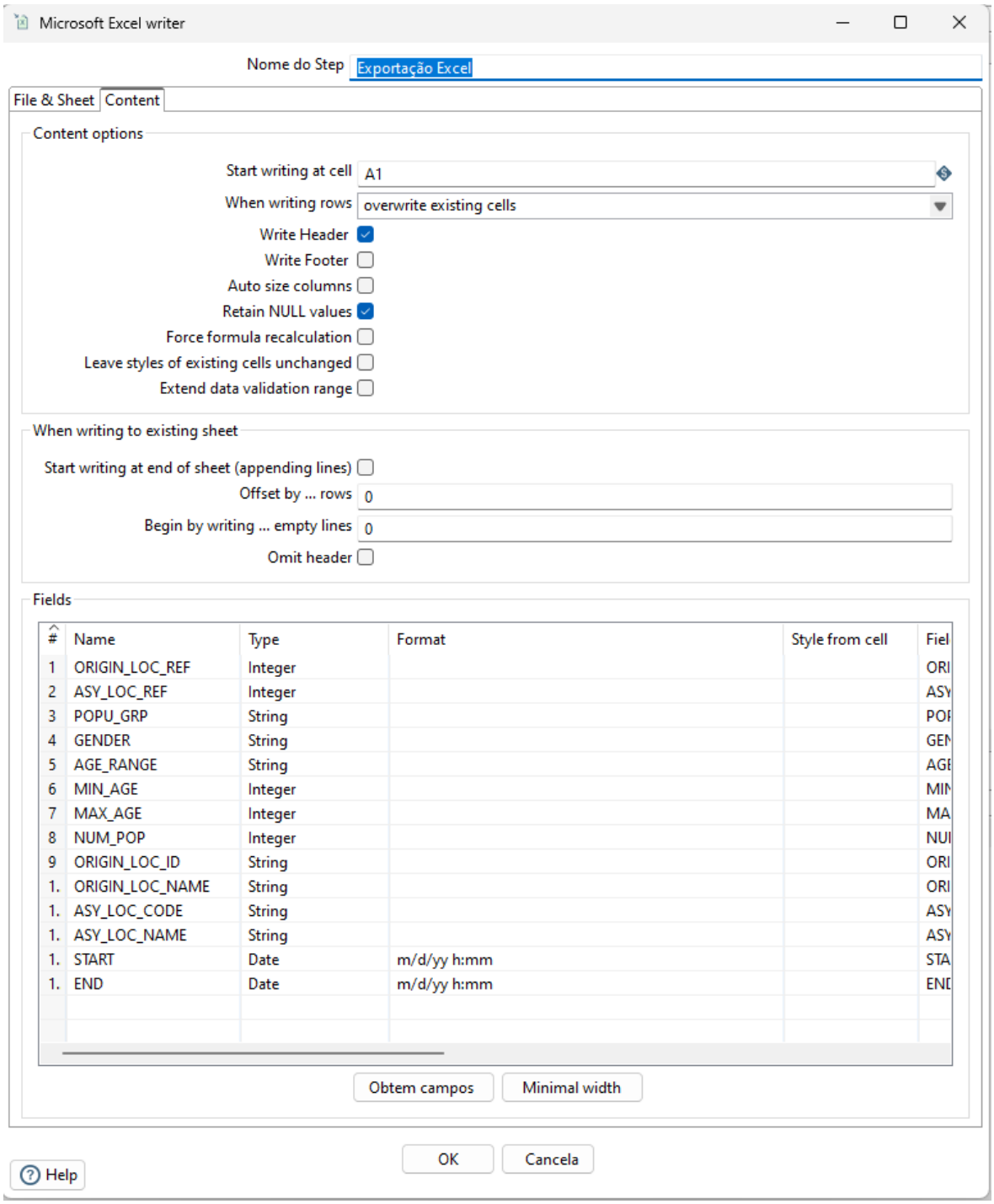

Figura 18 - Mapeamento das colunas

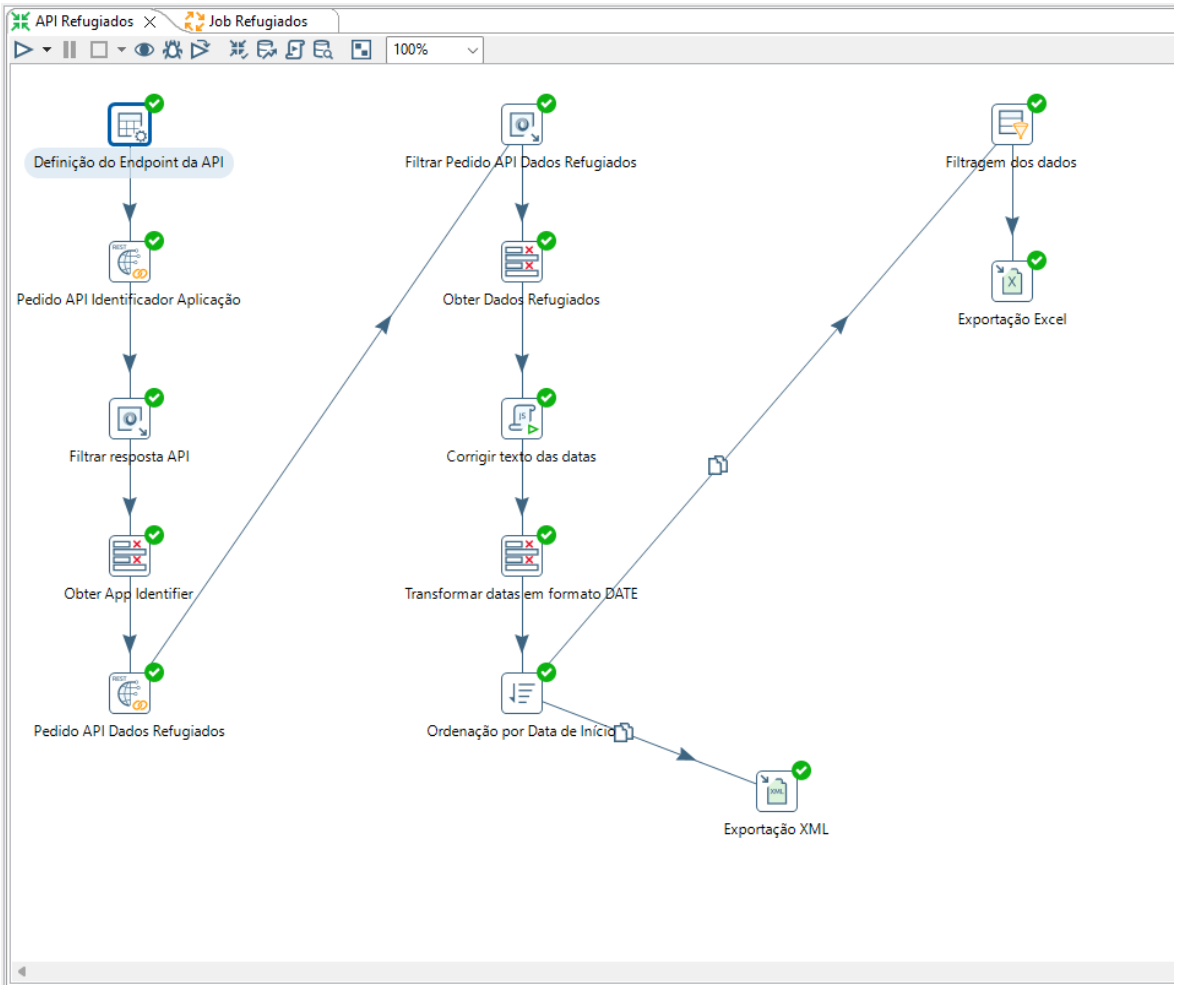

Figura 19 - Visão geral da transformação

3.2. Job

Foi criado um Job para permitir o envio do ficheiro Excel exportado na transformação para um email definido. Para isso foi incluída a execução da transformação dentro do Job, seguido de uma validação da existência do ficheiro exportado.

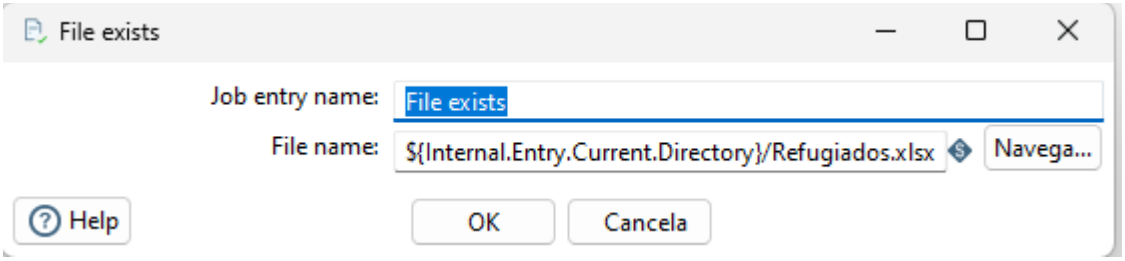

Figura 20 - Validação existência do ficheiro exportado

Caso esta validação seja positiva, ou seja, o ficheiro existe, é enviado então o email com as configurações definidas.

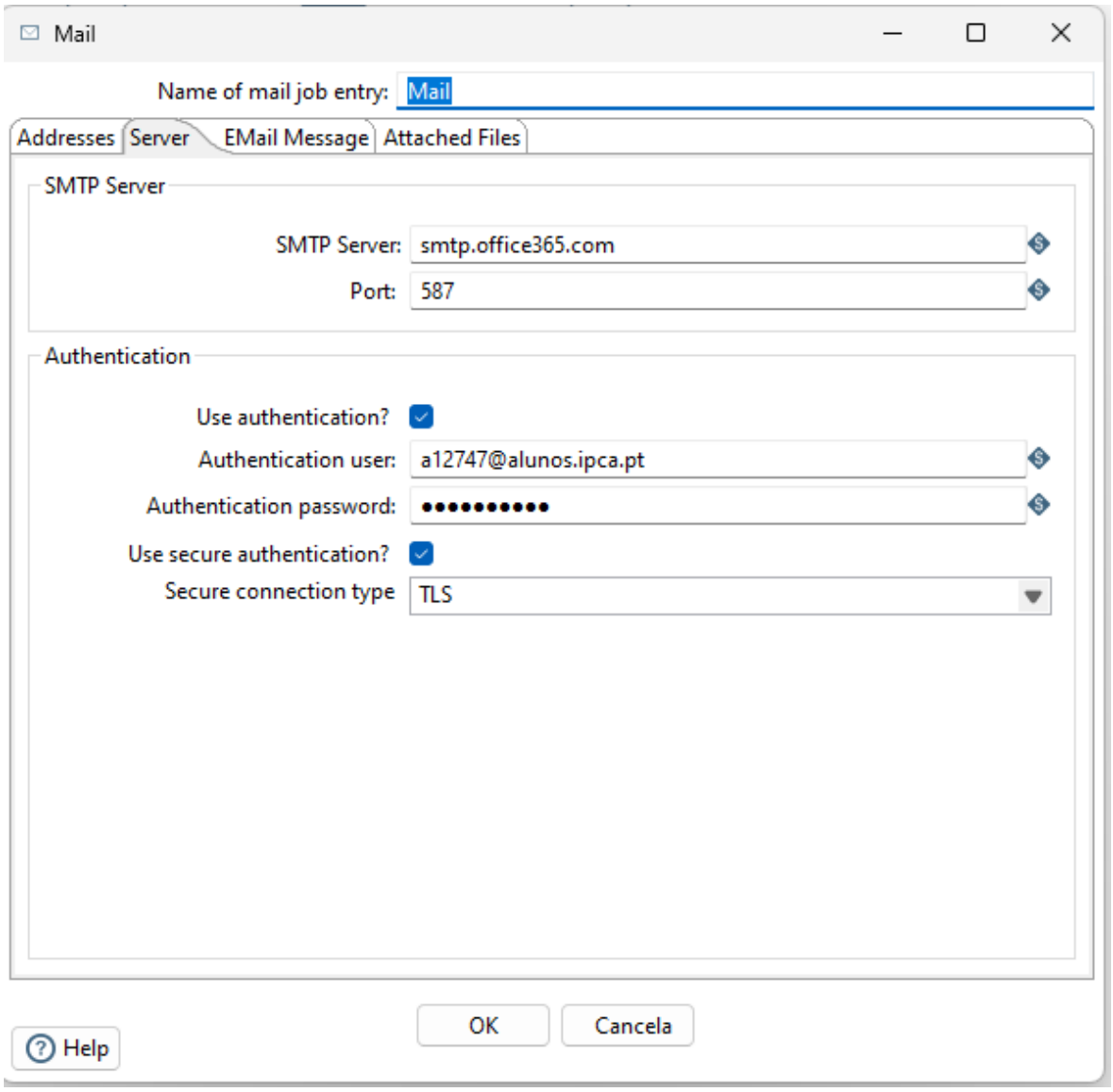

Figura 21 - Configuração do email

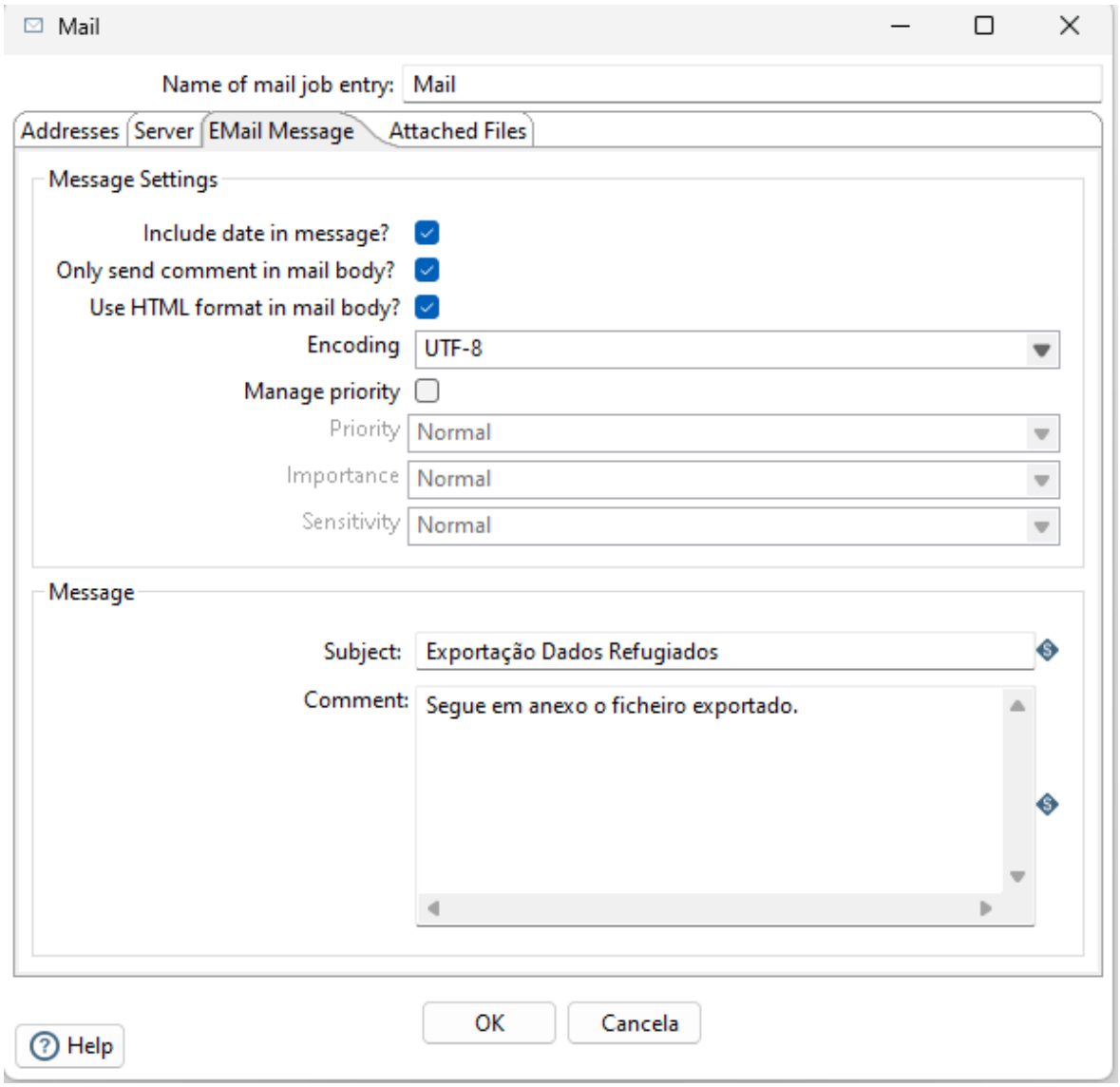

Figura 22 - Assunto e Mensagem a enviar no email

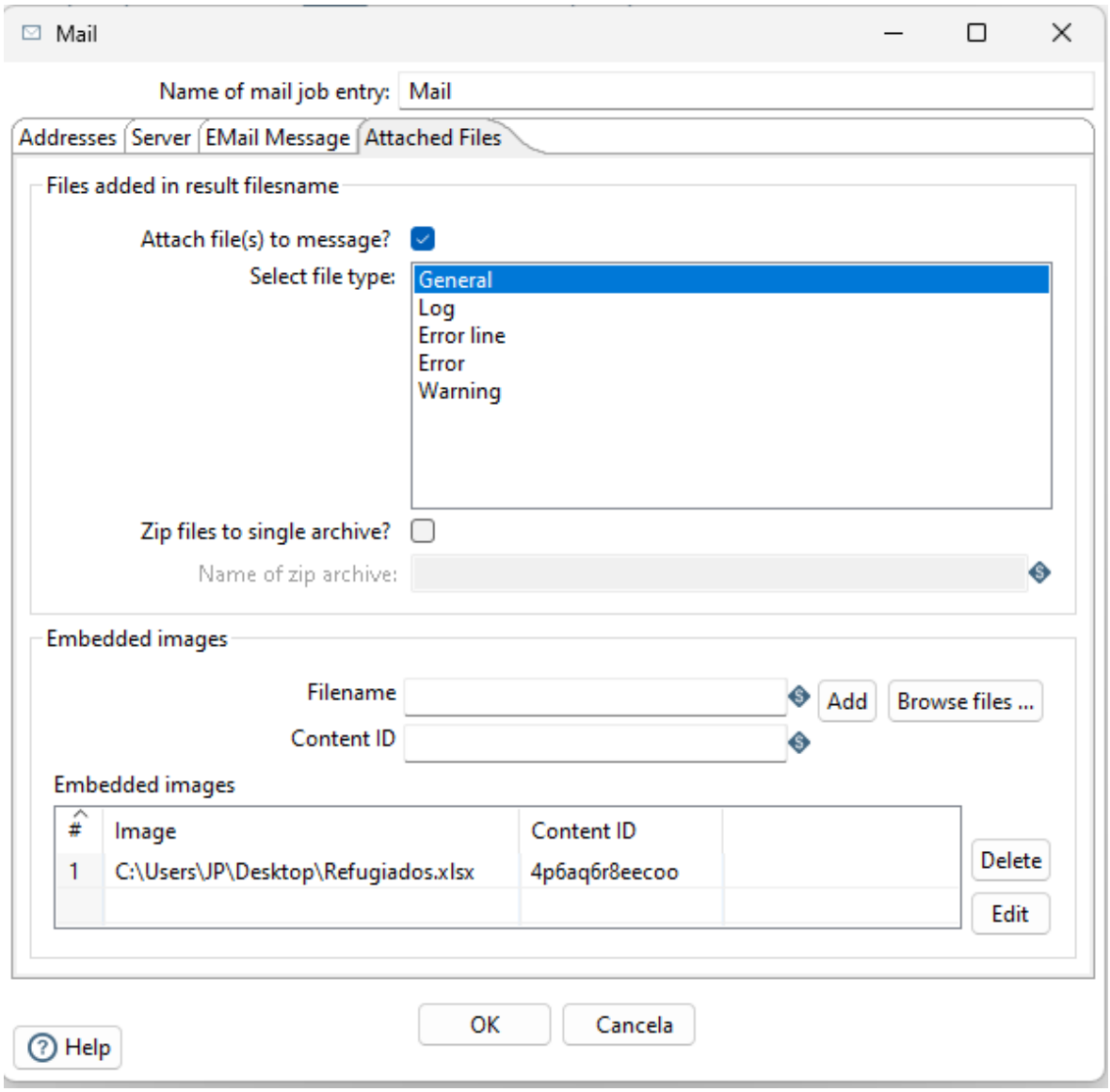

Figura 23 - Definição de ficheiro a anexar no email

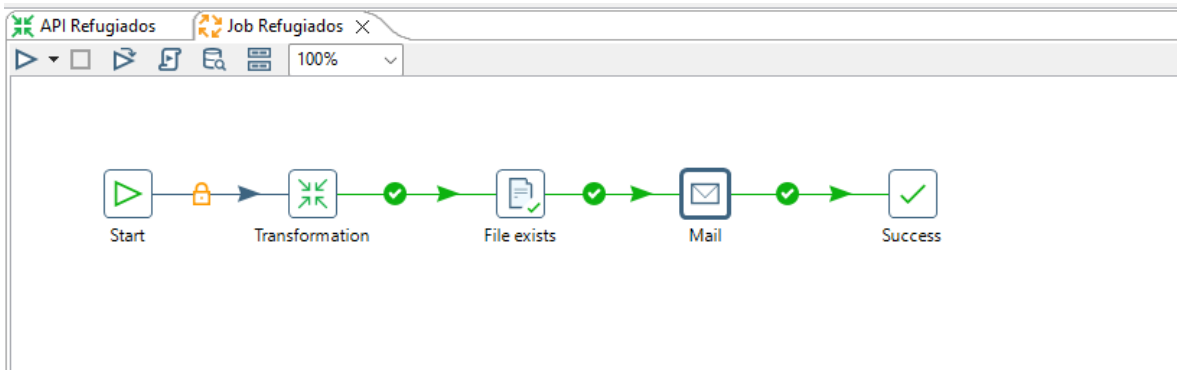

Figura 24 - Visão geral do Job

4. Vídeo

Deixo abaixo o código QR gerado com o link para o vídeo da execução do processo de transformação dos dados.

5. Conclusão

Este trabalho demonstrou que as ferramentas de ETL (Extract, Transform, Load) oferecem uma vasta gama de recursos para a integração, transformação e análise de dados. A experiência adquirida ao explorar essas ferramentas foi bastante enriquecedora, permitindo compreender diversas funcionalidades que se destacam pela simplicidade e versatilidade.

Além disso, a capacidade de se integrar com outras ferramentas, como bases de dados e APIs, e de extrair dados de múltiplas fontes, transformando-os conforme as necessidades específicas e carregando-os nos destinos desejados, representa um grande diferencial.

Os conhecimentos adquiridos nesta disciplina não apenas ampliam a compreensão dos processos ETL, mas também podem ser aplicados diretamente no contexto profissional, facilitando a resolução de desafios relacionados ao tratamento de dados.

Em resumo, as ferramentas de ETL são uma excelente escolha para análise e tratamento de dados, oferecendo vantagens significativas para seus usuários e contribuindo para o desenvolvimento de habilidades práticas no mercado de trabalho.

6. Anexos

- Exemplo Exportação Excel
- Exemplo Exportação XML

7. Bibliografia

- **API**
  - https://hdx-hapi.readthedocs.io/en/latest/data_usage_guides/affected_people/#refugees-persons-of-concern
  - https://hapi.humdata.org/docs#/
- **Kettle**
  - https://docs.hitachivantara.com/r/en-us/pentaho-data-integration-and-analytics/10.1.x/mk-95pdia000/getting-started-with-pdi/pentaho-data-integration-pdi-tutorial/step-1-extract-and-load-data