ESCOLA SUPERIOR DE TECNOLOGIA IPCA

Processos de ETL

Relatório do trabalho da disciplina

Integração de Sistemas de Informação

Marcio Alexandre Martins Dias – a20757

Licenciatura em Engenharia Sistemas Informáticos (Pós-Laboral)

Barcelos, outubro 2024

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.

Marcio Alexandre Martins Dias – a20757

1 Índice

Índice de Figuras

Enquadramento

Este trabalho, desenvolvido no âmbito da disciplina de Integração de Sistemas de Informação (ISI), tem como objetivo a aplicação e experimentação de ferramentas de ETL (Extract, Transformation, and Load), essenciais para a integração e tratamento de dados em sistemas de informação. O foco principal está no uso do Microsoft SQL Server Integration Services (SSIS) como a ferramenta central para o desenvolvimento dos processos de ETL.

O projeto consiste no tratamento de dados relacionados com filmes, especificamente no formato JSON. Este ficheiro contém várias colunas, tais como o título, ano de lançamento, realizador, entre outras informações relevantes sobre cada filme. O trabalho envolveu a extração dos dados, a sua transformação e o carregamento para um ambiente de armazenamento adequado, o SQL.

No processo de transformação, foram realizadas operações de limpeza e filtragem dos dados, incluindo a eliminação de colunas irrelevantes para o estudo, como as colunas “url”, “screenplay by”, “story by”, “based on”, “narrated by”, “music by”, “cenematografic” e “edited by".

Estas etapas foram fundamentais para otimizar o conjunto de dados, concentrando a análise nas informações mais pertinentes. Foram também exploradas funcionalidades avançadas do SSIS, como a configuração de tarefas de transformação e o uso de expressões condicionais para efetuar as correções e ajustes necessários.

Este projeto tem como objetivo demonstrar a capacidade de integrar dados provenientes de diferentes fontes, bem como evidenciar a eficiência e flexibilidade proporcionadas pelo SSIS no tratamento e transformação de grandes volumes de dados, facilitando o processo de análise e a tomada de decisões subsequentes.

Problema

Atualmente, a quantidade de dados disponíveis sobre filmes está em constante crescimento, incluindo informações como títulos, anos de lançamento, realizadores, entre outros detalhes. No entanto, estes dados encontram-se frequentemente dispersos em diversos formatos, como JSON e CSV, o que dificulta a sua recolha e análise de forma eficiente.

Uma grande parte destes dados não estão preparados para análise direta, necessitando de passar por processos de limpeza e transformação para uma consulta facilitada, colunas desnecessárias ou erros nos dados em bruto dificultam a sua utilização eficaz. Para permitir uma análise adequada, é essencial organizar e filtrar os dados, de modo a garantir que apenas as informações relevantes são guardadas.

O problema identificado neste projeto reside, assim, na necessidade de realizar a limpeza e organização dos dados para uma análise posterior, e transformar os dados do ficheiro JSON em dados organizados e estruturados. Após esse processo, os dados são integrados numa base de dados SQL, o que facilita a sua gestão e permite consultas mais eficientes, assegurando a consistência e qualidade dos dados.

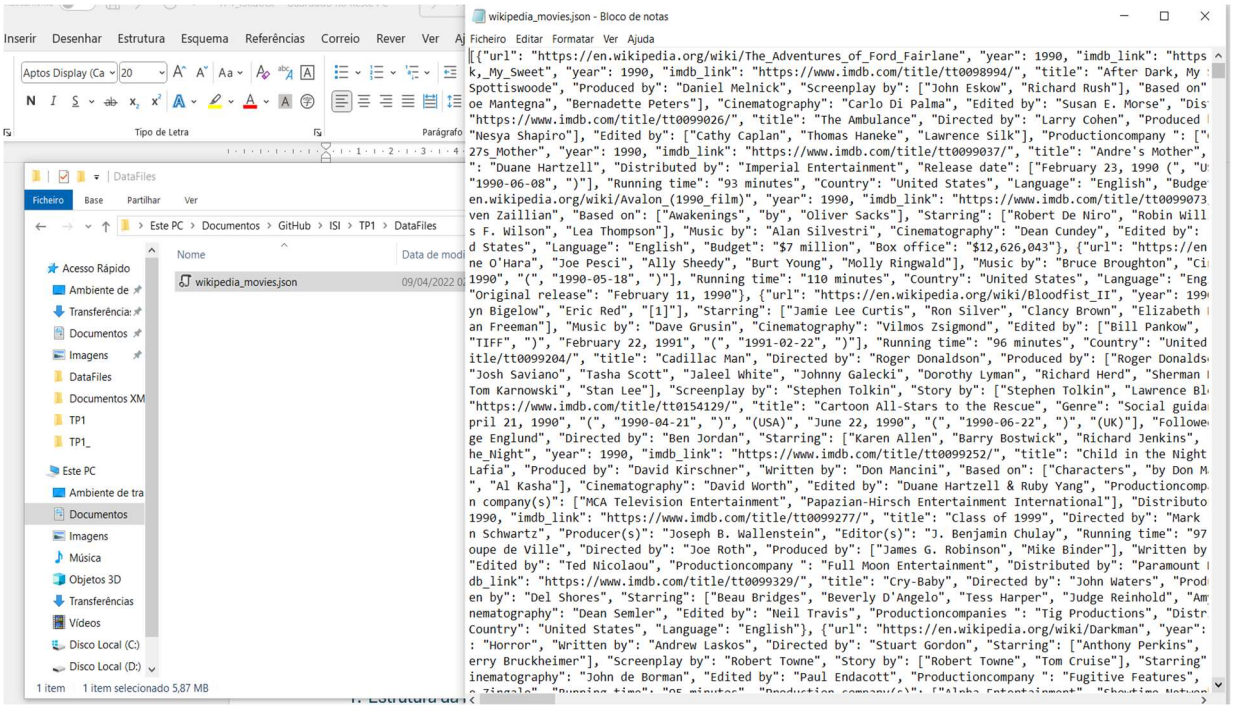

Figura 1 - Ficheiro Json Original

1. Estratégia Utilizada

1.1 Operadores e Processos envolvidos

Para alcançar os objetivos de limpeza, organização e integração dos dados, será utilizado o **Microsoft SQL Server Integration Services (SSIS)** como a principal ferramenta de ETL. No decorrer deste processo, diversos operadores e processos serão aplicados, divididos nas três fases principais: **Extração**, **Transformação** e **Carregamento**.

Extração:

- O processo de extração será responsável por recolher os dados de ficheiros no formato **JSON**. No SSIS, será utilizado o componente **JSON Source** para importar estes ficheiros e convertê-los num formato que possa ser processado nas etapas seguintes.
- Este operador irá mapear os diferentes campos presentes no ficheiro JSON (como [title], [year], [Directed by], entre outros), preparando os dados para a transformação.

Transformação:

Após a extração, serão aplicados operadores de transformação para limpar e organizar os dados:

- **Derived Column**: Este operador será utilizado para criar ou modificar colunas com base em expressões definidas. Neste trabalho foi utilizado para converter os valores da tabela [Relase Date] em string.
- **Conditional Split**: Será utilizado para filtrar e separar os dados com base em condições específicas, como a eliminação de registos com valores inválidos ou incompletos. Neste caso foram eliminados os dados das colunas com valores "NULL".
- **Script Component**: Este componente será fundamental para operações personalizadas que não podem ser realizadas pelos operadores padrão do SSIS. O **Script Component** permite a escrita de código em C# para realizar transformações avançadas, neste trabalho foi escrito código para implementar o REGEX e transformar os dados da tabela [Relase Date].

Carregamento:

- A fase final consiste no carregamento dos dados transformados para uma base de dados SQL, utilizando o componente **OLE DB Destination** no SSIS.
- Este processo garante que os dados tratados e limpos são armazenados de forma estruturada numa base de dados, prontos para serem consultados e analisados posteriormente. Esta estrutura facilita o acesso eficiente aos dados para futuras análises ou consultas SQL.

2 Transformações

Ao utilizar o **Microsoft SQL Server Integration Services (SSIS)** para o desenvolvimento de processos ETL, é essencial ter em conta a sua estrutura de construção. Cada processo global no SSIS é designado por um **package/job**, dentro de cada **package** são executadas as diversas tarefas e processos de **Transformação**, seguindo uma lógica estruturada para garantir a correta execução do fluxo de ETL.

2.1 Fluxo de controlo

No **Microsoft SQL Server Integration Services (SSIS)**, o **fluxo de controlo** (Control Flow) refere-se à parte do package que organiza e gere a sequência de execução das várias tarefas e subprocessos. Ele define a lógica de execução e a ordem em que as operações são realizadas dentro do package, funcionando como um "roteiro" que coordena o processamento de dados.

O fluxo de controlo é essencial para a coordenação do processo ETL, pois ele:

- Determina a sequência e a lógica de execução das tarefas.
- Garante que as operações sejam realizadas na ordem correta, e que erros ou condições específicas sejam geridos adequadamente.
- Permite a reutilização de código e a organização do processo ETL em blocos lógicos de tarefas.

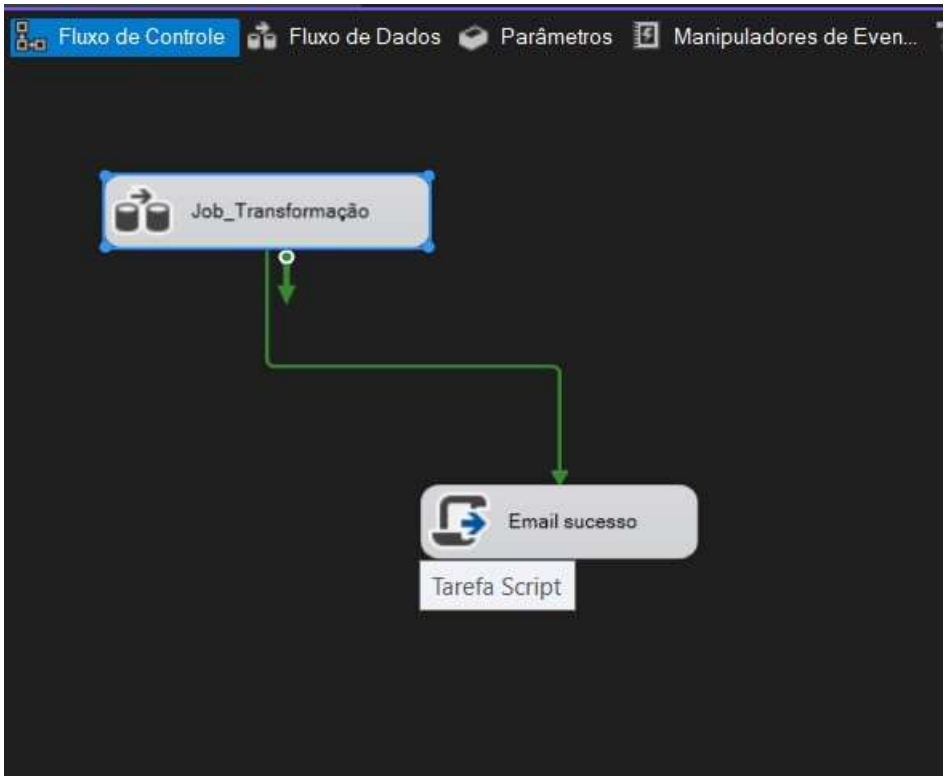

Figura 2 - Fluxo de controle

2.2 Fluxo de Dados

O fluxo de dados é onde a maior parte das operações de processamento ocorrem no SSIS. Ele é composto por um pipeline que liga as **fontes** de dados às **transformações** e, em seguida, aos **destinos**. A ordem em que as transformações são aplicadas é crítica, pois os dados fluem de uma operação para a outra de forma sequencial.

O **fluxo de dados** trabalha em conjunto com o **fluxo de controlo**, que determina quando e como o fluxo de dados será executado, coordenando-o com outras operações no package SSIS.

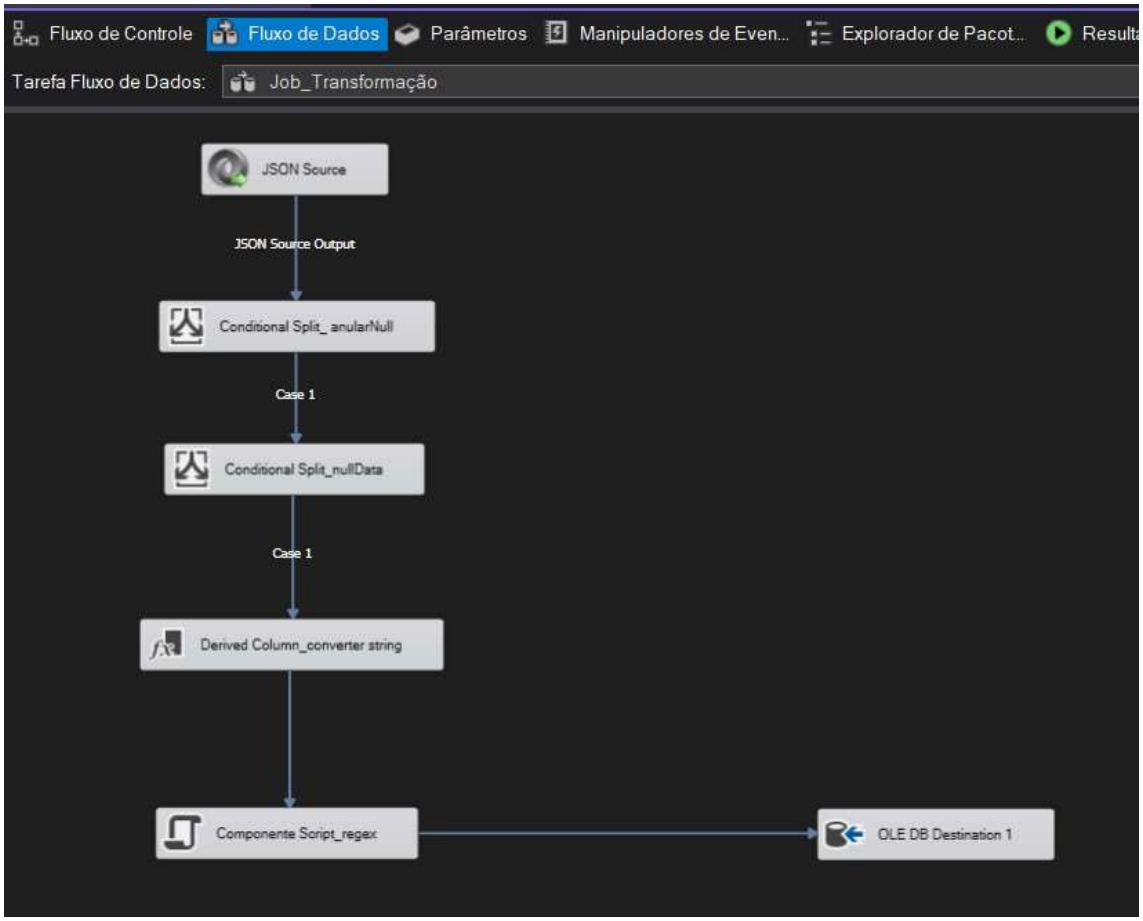

Figura 3 - Fluxo de Dados

2.2.1 Etapas do Fluxo de Dados

1. **Extração**:

   Os dados são extraídos das suas origens (por exemplo, um ficheiro JSON).

Utilizei a componente **JSON Source** para importar estes ficheiros e convertê-los num formato que possa ser processado nas etapas seguintes.

Figura 4 - JsonSource

Configurei o caminho onde o ficheiro se encontrava para que o operador dar início a extração de dados.

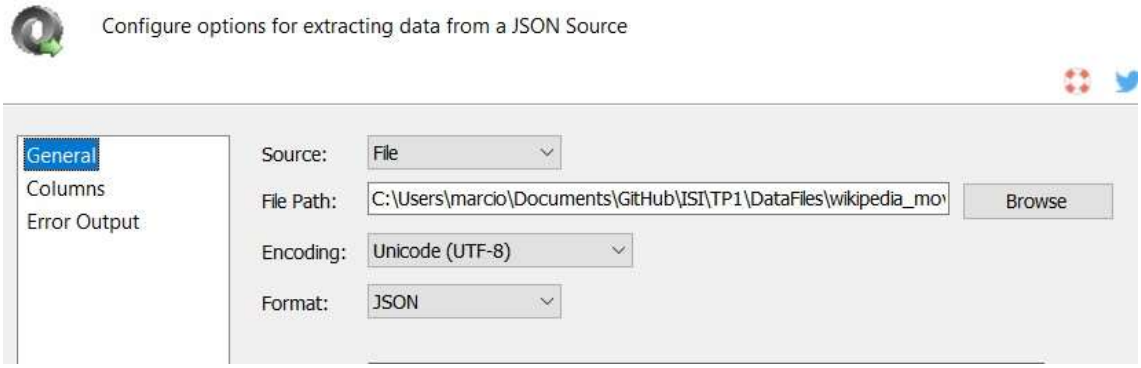

Figura 5 - CaminhoFicheiroJson

Foram definidas as colunas de entrada e as colunas de saída para serem transformadas daqui para a frente.

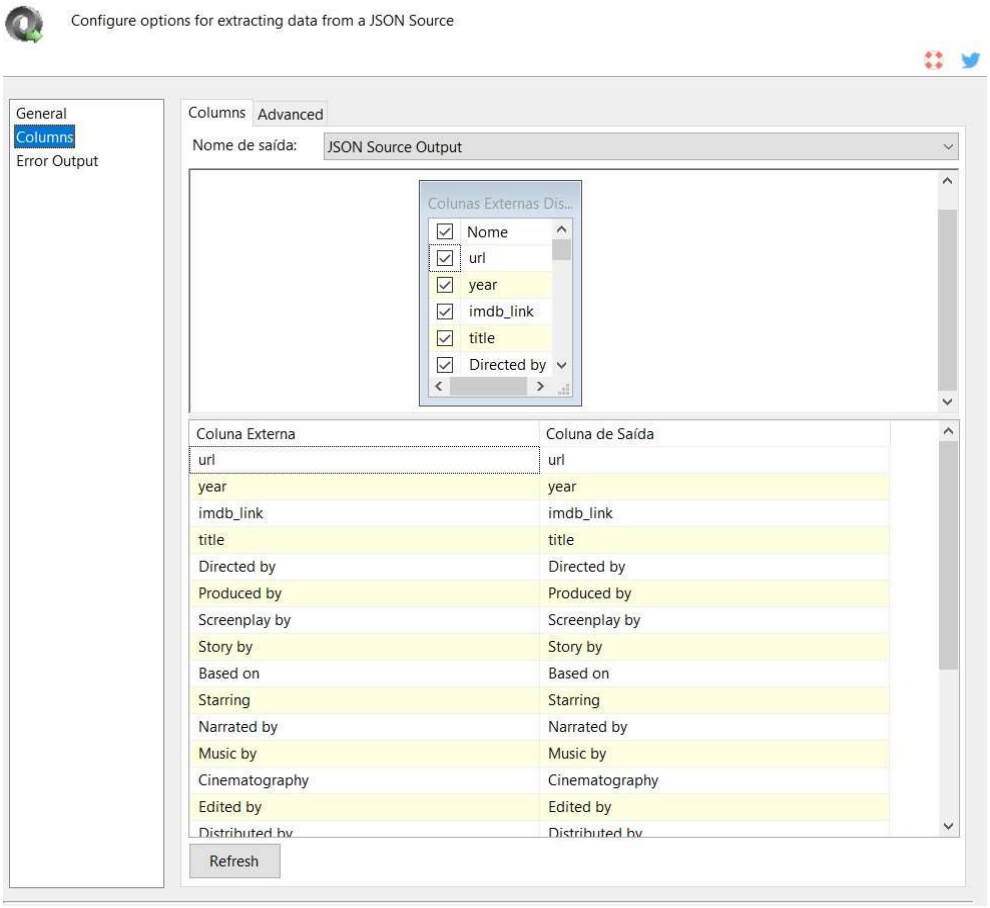

Figura 6 - Colunasjson

2. **Transformação**:

   À medida que os dados percorrem o fluxo, passam por uma série de transformações.

   Neste ponto, pode-se aplicar validação de dados, agregações, cálculos e limpeza

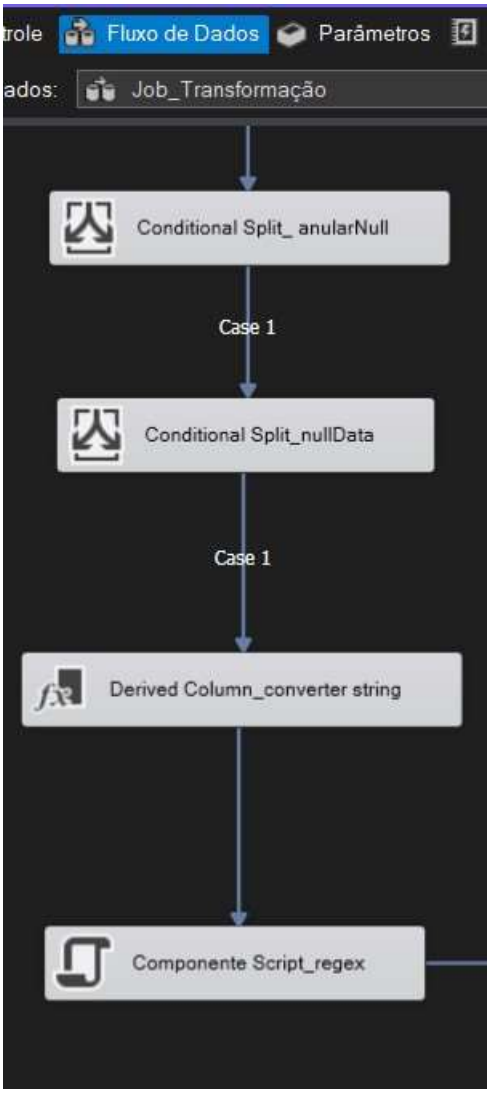

Figura 7 - Varias transformações

Conditional Split

No processo de ETL, utilizei o operador **Conditional Split** para filtrar todas as linhas que continham valores **Null**.

Este operador foi configurado para verificar a presença de valores vazios em colunas específicas e, ao encontrar registos com dados nulos, esses foram automaticamente excluídos do fluxo de dados.

Desta forma, garantiu-se que apenas dados completos e válidos prosseguissem para as etapas seguintes do processo.

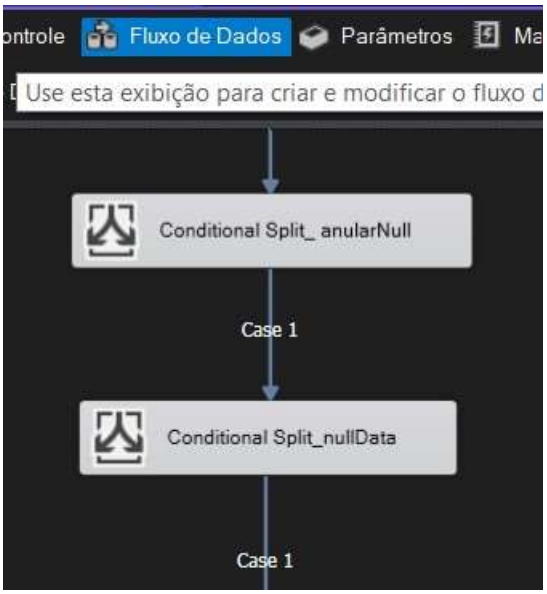

Figura 8 - Conditional Split

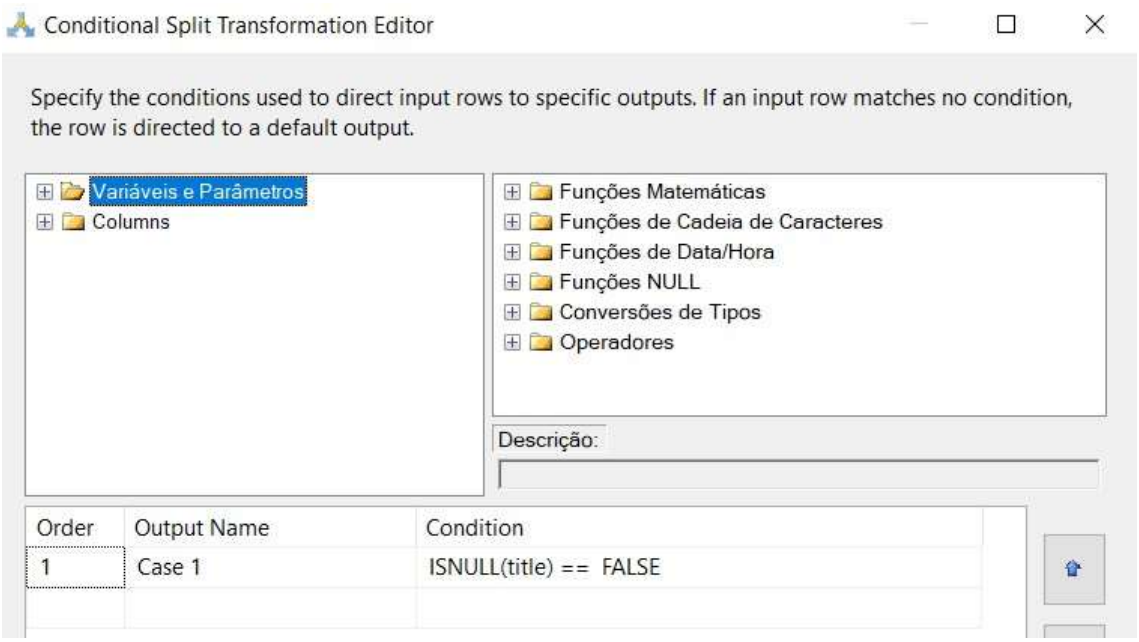

Figura 9 - Eliminar linhas em branco(NULL)

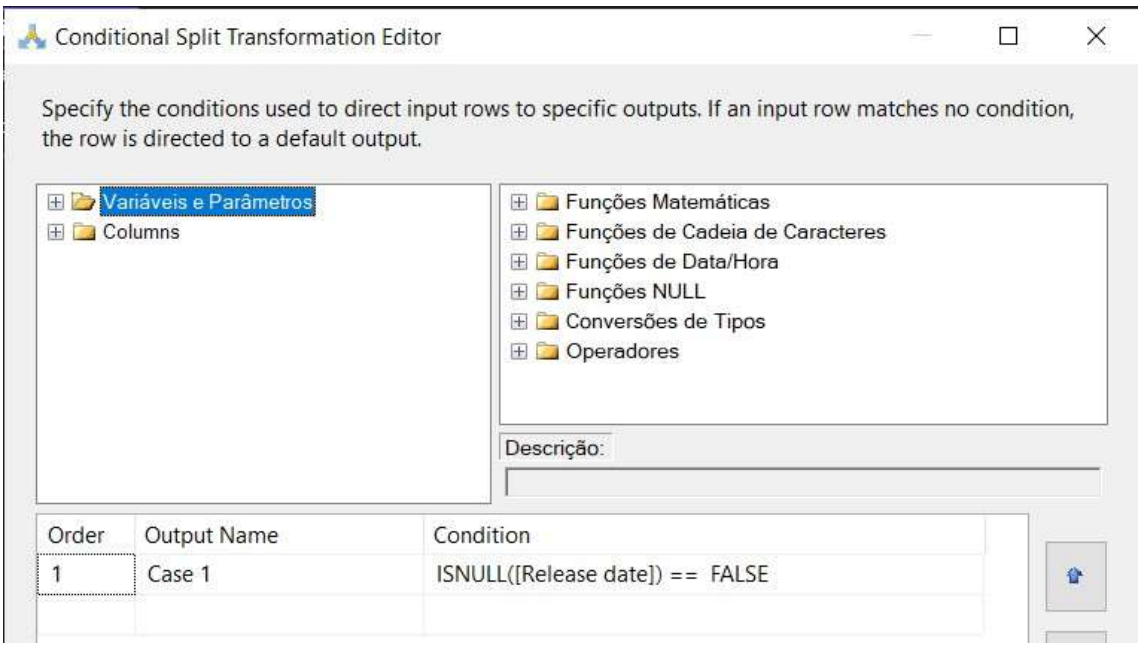

Figura 10 - Eliminar linhas em branco(NULL) da coluna Relase Date

Derived Column

O operador **Derived Column** foi utilizado para converter a coluna **"release date"** (data de lançamento) num formato de **string**.

Esta transformação foi necessária para padronizar o formato da data e permitir uma manipulação mais eficiente nos próximos passos do ETL.

A conversão permitiu que a data fosse tratada como texto, o que facilitou a sua utilização em operações de concatenação e exibições específicas.

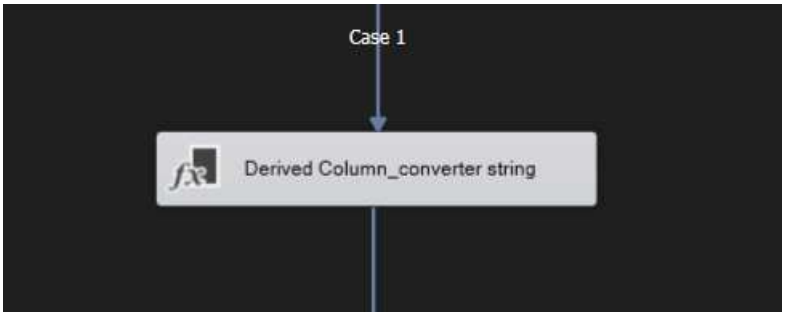

Figura 11 - Derived Column

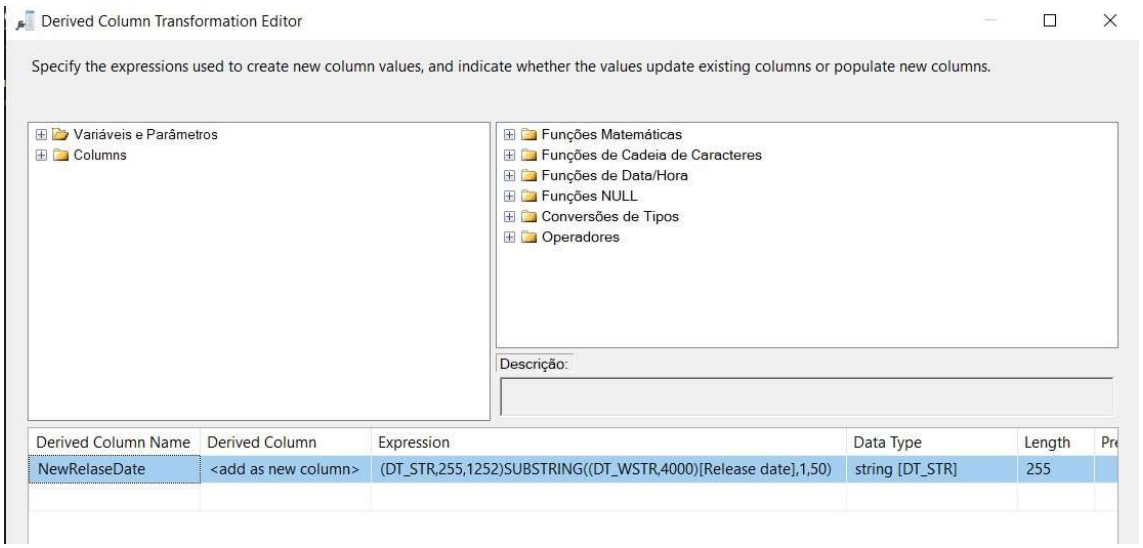

Figura 12 - Passagem para string os dados Relase Date

Script Component

O **Script Component** foi utilizado para manipulação avançada dos dados através de expressões **Regex**. O código personalizado, escrito em C#, permitiu aplicar regras de validação e formatação aos dados, ajustando e validando padrões de texto complexos que não podiam ser facilmente resolvidos com os componentes padrão do SSIS.

Este componente foi essencial para garantir que os dados seguissem os formatos requeridos e corrigir entradas incorretas ou mal formatadas.

A transformação foi efetuada na coluna [Relase Date], os dados estavam desconfigurados e com datas incompletas e o REGEX efetuou a manipulação ficando os dados corrigidos.

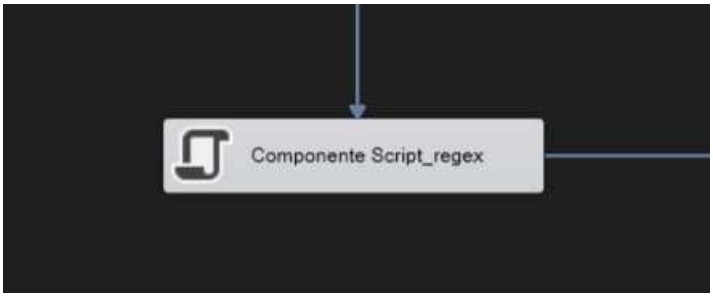

Figura 13 - Component Script

```
/// <param name="Row">The row that is currently passing through the component</param>
public override void Entrada0_ProcessInputRow(Entrada0Buffer Row)
{
    string dia = @"\d{2}";
    string ano = @"\d{4}";
    string mes = @"([a-zA-Z]+)";
    string releaseDateString = Row.NewRelaseDate;

    if (Regex.IsMatch(releaseDateString, ano))
    {
        Row.ano = int.Parse(Regex.Match(releaseDateString, ano).Value);
    }
    else { Row.ano = 0000; }
    if(Regex.IsMatch(releaseDateString, dia))
    {
        Row.dia = int.Parse(Regex.Match(releaseDateString, dia).Value);
    }
    else { Row.ano = 00; }

    if(Regex.IsMatch(releaseDateString, mes))
    {
        Row.mes = Regex.Match(releaseDateString, mes).Value;
    }else { Row.mes = "indefenido";}
```

Figura 14 - Codigo REGEX (C#)

3. **Carregamento**:

   Os dados processados são enviados para o destino final, como uma base de dados ou ficheiro.

OLE DB Destination

Para concluir o processo, utilizei o componente **OLE DB Destination** para ligar ao **SQL Server** e criar uma nova tabela Movie.Final onde foram armazenados os dados já limpos e corrigidos.

Este componente permitiu que o fluxo de dados fosse carregado diretamente na base de dados, garantindo que a informação estivesse disponível e organizada para futuras consultas ou análises.

Figura 15 - OLE DB Destination - Ligar ao SQL

```
CREATE TABLE [dbo].[Movies_Final](
    [year] [bigint] NULL,
    [imdb_link] [nvarchar](256) NULL,
    [title] [nvarchar](256) NULL,
    [Directed by] [nvarchar](256) NULL,
    [Produced by] [nvarchar](max) NULL,
    [Starring] [nvarchar](max) NULL,
    [Distributed by] [nvarchar](256) NULL,
    [Running time] [nvarchar](256) NULL,
    [Country] [nvarchar](256) NULL,
    [Language] [nvarchar](256) NULL,
    [Budget] [nvarchar](256) NULL,
    [Box office] [nvarchar](256) NULL,
    [Dia] [int] NULL,
    [Mes] [varchar](max) NULL,
    [Ano] [int] NULL

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Figura 16 - Criar nova tabela no SQL

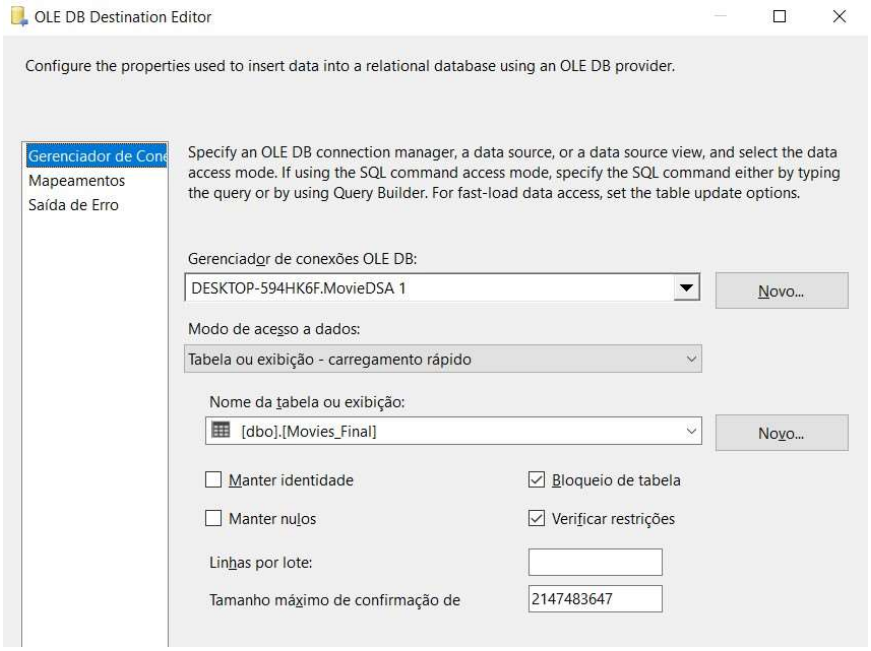

Figura 17 - Ligação a BD e a tabela Movies_Final

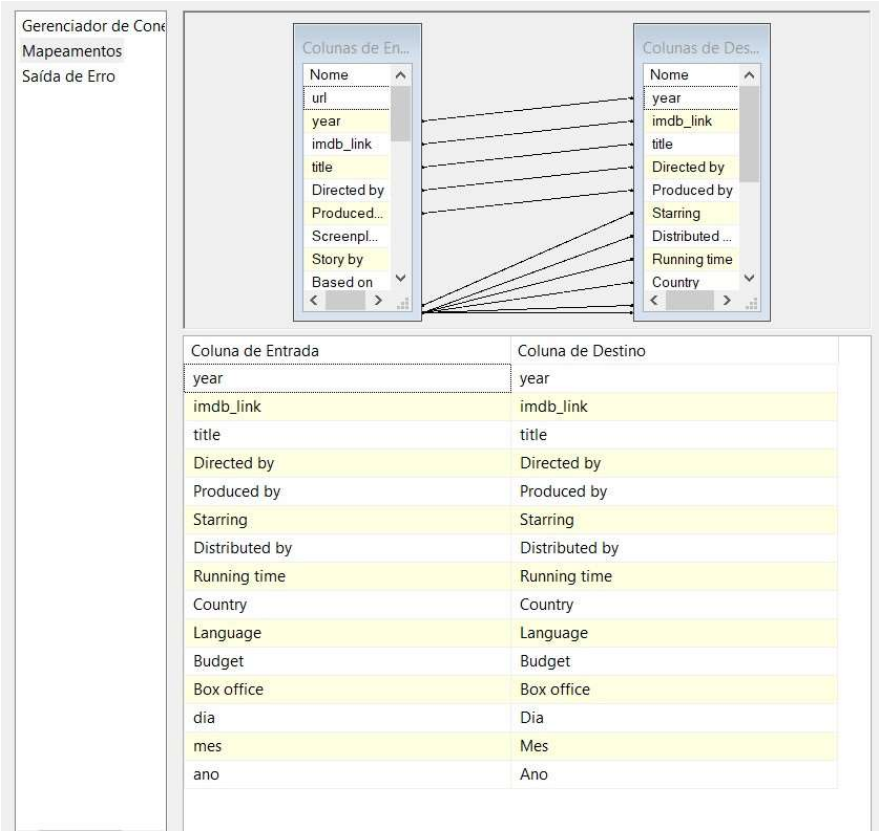

Figura 18 - Mapeamento colunas final

2.3 Outras implementações

Tarefa Script

Adicionalmente, configurei uma **tarefa Script** para enviar um e-mail automático de sucesso, confirmando que todo o processo de ETL foi executado corretamente.

Caso ocorra algum erro durante o processo, foi criada uma tarefa alternativa no "Manipuladores de evento" que também utiliza o componente **Script**, responsável por enviar um e-mail de erro, alertando sobre o problema ocorrido.

Com esta configuração, é possível monitorizar o fluxo de ETL e agir rapidamente caso seja identificado algum erro, garantindo a fiabilidade do processo.

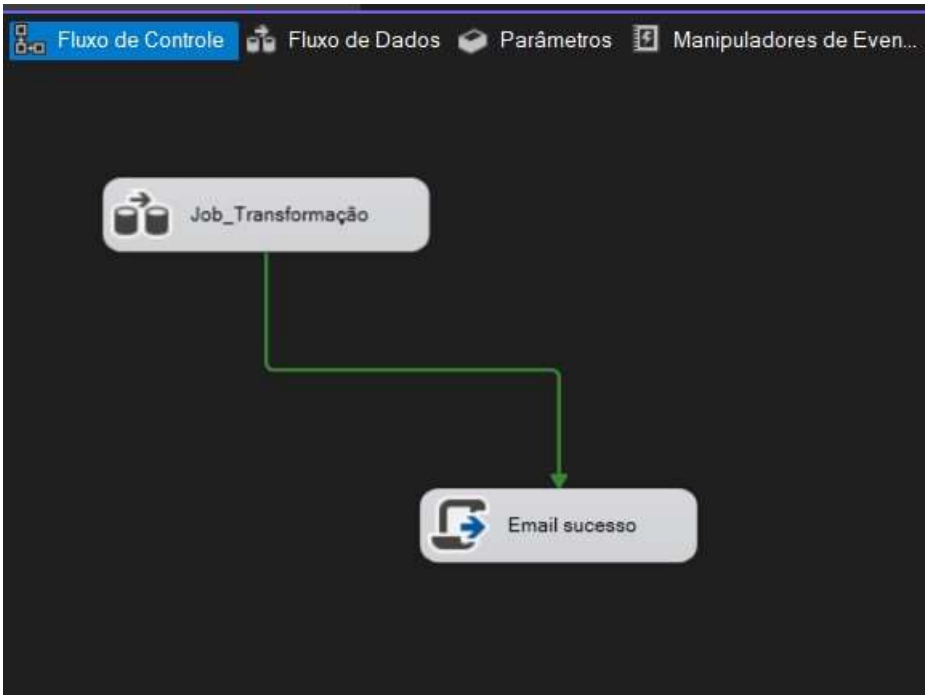

Figura 19 - Email Sucesso

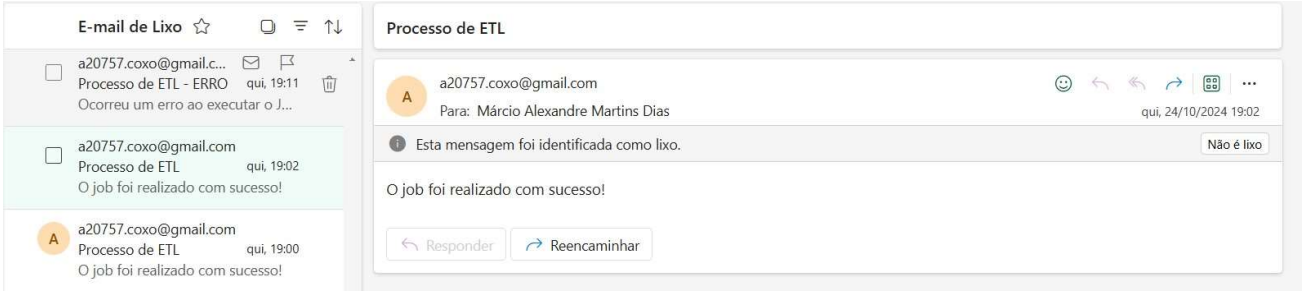

Figura 20 - Email Sucesso enviado

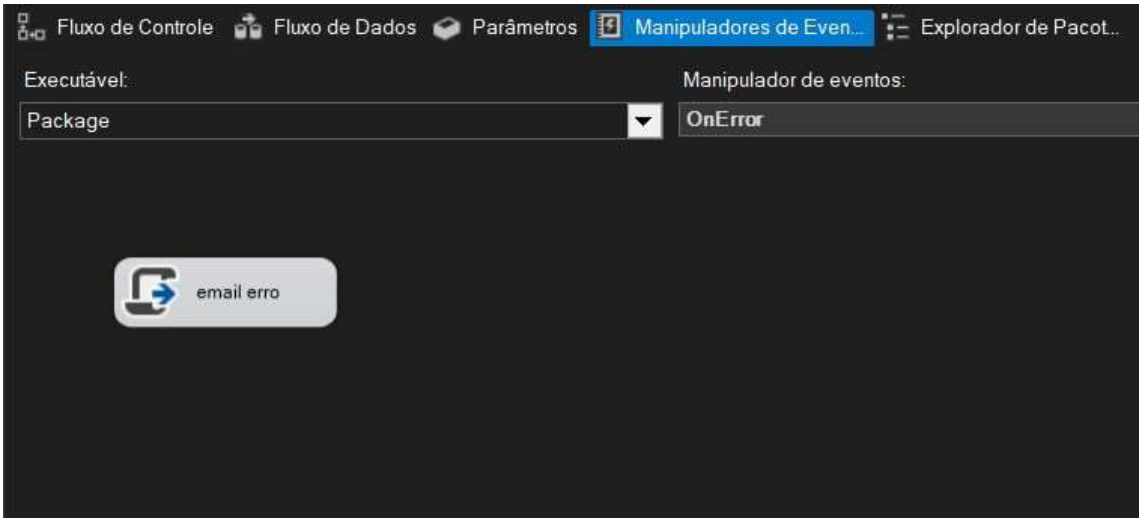

Figura 21 - Email Erro

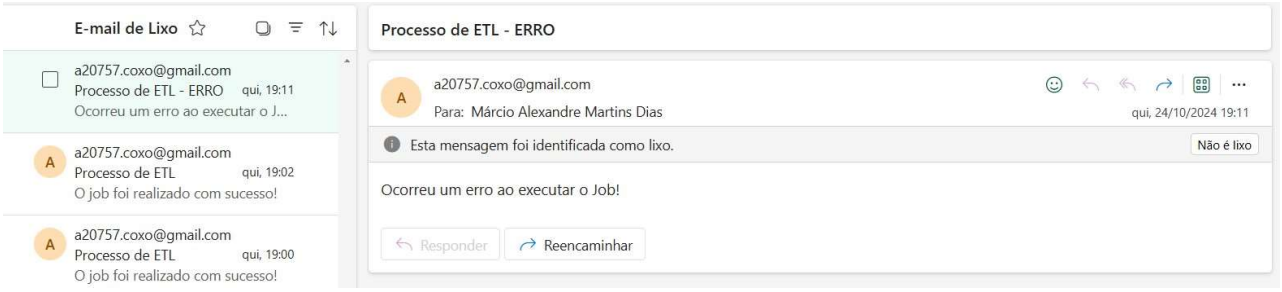

Figura 22 - Email Erro Enviado

```
public void Main()
{
    // Configurações do cliente SMTP
    string smtpHost = Dts.Variables["$Project::SMTPHost"].Value.ToString();
    int smtpPort = int.Parse(Dts.Variables["$Project::SMTPPort"].Value.ToString());
    string smtpUser = Dts.Variables["$Project::SMTPUser"].Value.ToString();
    string smtpKey = Dts.Variables["$Project::SMTPKey"].Value.ToString();
    string smtpTo = Dts.Variables["$Project::SMTPTo"].Value.ToString();

    // Configurações do cliente SMTP
    SmtpClient client = new SmtpClient(smtpHost, smtpPort)
    {
        UseDefaultCredentials = false,
        Credentials = new NetworkCredential(smtpUser, smtpKey),
        EnableSsl = true
    };

    // Criando a mensagem de e-mail
    MailMessage mailMessage = new MailMessage
    {
        From = new MailAddress(smtpUser),
        Subject = "Processo de ETL",
        Body = "O job foi realizado com sucesso!",
        IsBodyHtml = true // Se você quiser enviar e-mails em HTML
    };

    mailMessage.To.Add(smtpTo);

    // Enviando o e-mail
    client.Send(mailMessage);
}
```

Figura 23 - Código Configurar email (C#)

Parâmetros globais

No processo de construção do ETL, configurei **parâmetros globais** no SSIS para definir as configurações de e-mail, permitindo maior flexibilidade e controlo nas notificações automáticas de sucesso ou falha do processo. Estes parâmetros globais incluem informações como o endereço do remetente, o endereço do destinatário, o servidor SMTP, a porta e o assunto do e-mail.

Ao utilizar parâmetros globais, todas as configurações de e-mail podem ser facilmente ajustadas sem a necessidade de modificar o código interno da tarefa.

Este método centralizado facilita a manutenção e a adaptação das configurações de e-mail a diferentes ambientes, permitindo ainda que as mesmas configurações sejam reutilizadas em outras tarefas de e-mail dentro do mesmo projeto. Desta forma, garante-se uma estrutura organizada e eficiente na gestão de notificações do ETL.

3 Vídeo

Para demonstrar o funcionamento do ETL, fiz um vídeo que mostra o fluxo a correr no **SSIS**. No vídeo, apresento as várias etapas do processo, desde a extração inicial dos dados até à transformação e carregamento final na base de dados SQL.

É possível visualizar cada componente em ação, incluindo o **Conditional Split**, o **Derived Column**, o **Script Component**, e o **OLE DB Destination**, bem como as configurações de e-mail configuradas para notificações de sucesso e falha.

O vídeo explica todas as transformações.

Figura 24 - QRCode

4 Conclusão

Este projeto permitiu aplicar e consolidar conhecimentos em **ETL** (Extração, Transformação e Carregamento) utilizando o **Microsoft SQL Server Integration Services (SSIS)**. Através da criação de um fluxo de trabalho robusto e bem estruturado, foi possível extrair dados de várias fontes, transformá-los e carregá-los numa base de dados SQL com precisão e eficiência.

No decorrer do processo, foram implementados componentes chave como o **Conditional Split** para filtragem de dados, o **Derived Column** para padronização de formatos, o **Script Component** para manipulação avançada com expressões Regex, e o **OLE DB Destination** para a inserção dos dados finais numa tabela SQL. A utilização de **parâmetros globais** permitiu uma configuração centralizada para o envio de notificações por e-mail, melhorando a monitorização e resposta rápida em caso de falhas.

Além disso, a gravação de um vídeo demonstrativo ajudou a visualizar o fluxo do processo em tempo real, destacando cada etapa e transformação. Esta abordagem foi fundamental para assegurar a transparência e a eficácia do ETL, cumprindo com os objetivos do projeto de organizar, limpar e estruturar os dados de forma clara e acessível.

Em resumo, este trabalho proporcionou uma experiência prática valiosa na implementação de um sistema de integração de dados, demonstrando como as ferramentas de ETL podem transformar dados dispersos em informação útil e estruturada, e reforçando a importância da automação e monitorização contínua em processos de integração de dados.

5 Bibliografia

https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16

Cozyroc - https://www.cozyroc.com/ssis/json

https://www.wiseowl.co.uk/integration-services/videos/ssis-basics/