

# **Integração de Sistemas de Informação**

1ºTrabalho prático

Gonçalo Costa, N°26024

Docente: Óscar Ribeiro

## Índice

Integração de Sistemas de Informação.....	1
Introdução.....	4
Problema .....	5
Estratégia Utilizada .....	6
Softwares Utilizados .....	6
SQL Server .....	7
Transformações.....	9
Transformações Desenvolvidas .....	9
.....	9
Jobs.....	20
Jobs Desenvolvidos.....	20
Email .....	23
Node-RED .....	26
Vídeo demonstrativo .....	28
Conclusão.....	29
Bibliografia/Web grafia .....	30

## Índice de Figuras

Figura 1.1: Criação de Login no SQL.....	7
Figura 1.2: Conexão realizada no Pentaho .....	8
Figura 1.3: Query para criação da tabela no SQL .....	8
Figura 2: Transformação htmlAsAmbient .....	9
Figura 3: Transformação 1 .....	10
Figura 3.1: Passos de enviar dados para a base de dados.....	10
Figura 3.2: Filter Rows onde expressão regular .....	11
Figura 3.3: If field value is null Transformação 1 .....	11
Figura 3.4: If field value is null.....	11
Figura 3.5: Verificação se a música tem link_url .....	12
Figura 3.6: Verificação se a música tem mais de 1 artista.....	12
Figura 3.7: Renomeação dos campos com o Select Values.....	12
Figura 4: Transformação 2.....	13
Figura 4.1: Acesso a Base de dados pretendida e query SQL .....	13
Figura 4.2: Ordenação pelos BPM de forma decrescente .....	14
Figura 4.3: Script com o código .....	14
Figura 4.4: Regex Evaluation.....	15
Figura 4.5: Verificação se passou ou não .....	15
Figura 4.6: Verificação de Cantor.....	16
Figura 5: Transformação 3.....	17
Figura 5.1: Acesso a Base de dados pretendida e query SQL .....	17
Figura 5.2: Critério de Classificação quanto aos BPM's .....	18
Figura 5.3: Filtragens para as músicas de Velocidade Moderada .....	18
Figura 5.4: Chamada dos passos Sort Rows .....	19
Figura 5.5: Filtragem do Sort Rows para ordenar pelos BPM .....	19
Figura 6: Job 1.....	20
Figura 6.1: Eliminação do ficheiro XML.....	20
Figura 6.2: Evocação da Transformação.....	21
Figura 6.3: Log erro Execução da Transformação 1 .....	21
Figura 6.4: Verificações Necessárias .....	21
Figura 6.5: XSL Transformation .....	21
Figura 6.6: Transformation HTML_ambient.....	22
Figura 6.7: Envio de Email .....	22
Figura 6.8: Log de Sucesso! .....	22
Figura 7: Email que chegou à caixa de entrada do destinatário .....	23
Figura 7.1: Configuração de Email (destination e sender address) .....	24
Figura 7.2: Configuração de Email (server).....	24
Figura 7.3: Configuração de Email (mensagem de email).....	25
Figura 8: Esquema realizado no Node-RED .....	26
Figura 8.1: Chamada da function e conexão com SQL .....	26
Figura 8.2: Query SQL.....	27
Figura 8.3: Resultado do Esquema de Node-RED.....	27
Figura 9: QR Code.....	28

# Introdução

Com este trabalho para a unidade curricular de Integração de Sistemas de Informação (ISI) pretende-se focar a aplicação de ferramentas em processos de ETL (Extract, Transformation and Load), inerentes a processos de Integração de Sistemas de informação ao nível dos dados.

Pretende-se que sejam desenvolvidos processos de ETL que envolvam scripts próprias ou que recorram a ferramentas disponíveis como o Pentaho Kettle, Microsoft SQL Server Integration Services. Ferramentas complementares como node-RED, poderão também ser exploradas e integradas nos processos.

# Problema

A análise de dados de streaming de música, especialmente de plataformas como o Spotify, fornece insights valiosos sobre tendências de consumo, preferências dos utilizadores e desempenho de artistas. Contudo, a natureza volumosa e variada dos dados contidos em ficheiros CSV representa um desafio significativo para os analistas. Estes ficheiros não apenas incluem informações básicas, como nomes de artistas e faixas, mas também métricas complexas, como o número de streams, classificações e outros indicadores de desempenho.

Para abordar estas questões, um processo de ETL (Extract, Transform, Load) eficaz torna-se indispensável. Este processo não só ajudará a otimizar o fluxo de dados, mas também garantirá a qualidade e integridade da informação analisada.

A implementação de ferramentas de ETL apropriadas permitirá a automação de tarefas repetitivas, a redução de erros manuais e, em última instância, a capacidade de gerar relatórios e análises em tempo real.

Assim, este projeto visa desenvolver uma solução robusta que possa enfrentar estes desafios e facilitar a exploração e interpretação dos dados de streaming de música.

## Estratégia Utilizada

Neste projeto, a estratégia de ETL (Extract, Transform, Load) foi implementada utilizando o Pentaho Data Integration (PDI), para manipulação de dados.

A entrada de dados ocorre a partir de um ficheiro CSV, que contém todas as informações necessárias para a transformação. Durante esta fase inicial, são realizadas verificações para identificar se existem valores nulos. Nos casos em que isso acontece, são atribuídos valores padrão para assegurar a continuidade do processamento.

É igualmente importante verificar a legibilidade dos caracteres apresentados. Isso implica a deteção de caracteres especiais que possam causar erros na leitura do conteúdo dos ficheiros. Para contornar essas situações, outros caracteres ou expressões são assumidos, garantindo a integridade dos dados.

Ao longo de todas as transformações, diversas operações de filtragem e ordenação são aplicadas. Através destas operações, garantimos que apenas os valores relevantes sejam utilizados e que não existam anomalias nos ficheiros, como dados duplicados ou inconsistentes.

Finalmente, os dados tratados são armazenados num servidor SQL, permitindo a sua consulta e leitura para futuras transformações necessárias. Esta abordagem não só facilita a gestão dos dados, mas também assegura que as informações estejam organizadas e prontas para análises subsequentes.

## Softwares Utilizados

- Pentaho Kettle
- SQL Server
- Node-Red
- GitHub: <https://github.com/Goncalo04Costa/ISI-TP01-26024>

# SQL Server

O SQL server foi utilizado para guardar dados que irão ser utilizados em futuras operações.

Para conseguir utilizar conectar SQL como Pentaho, foi criado uma chave de acesso ao SQL Server

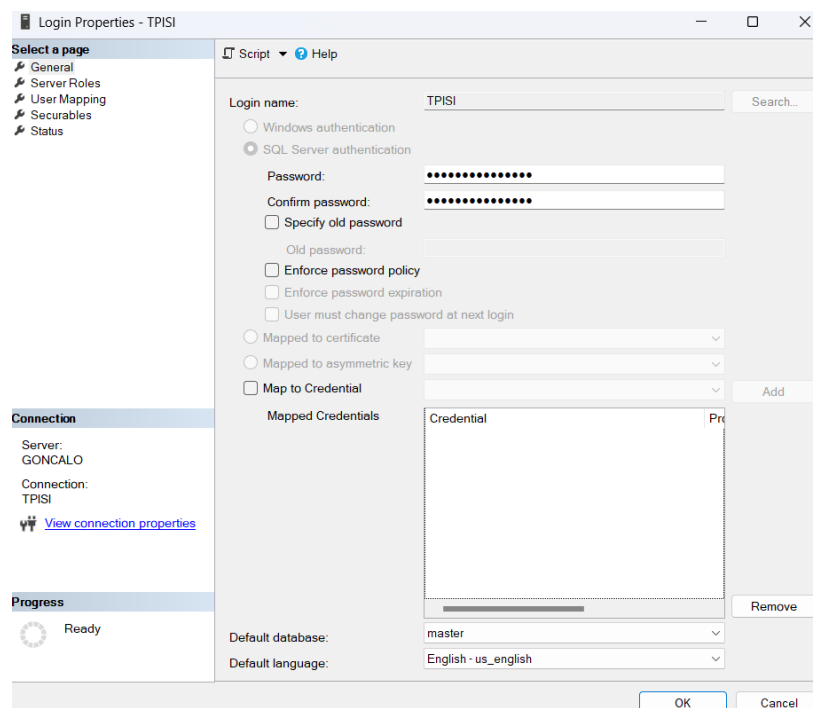


Figura 1.1: Criação de Login no SQL

Após isso, é efetuada a conexão entre o SQL Server com o Pentaho tendo em conta o Login que foi criado na imagem acima.

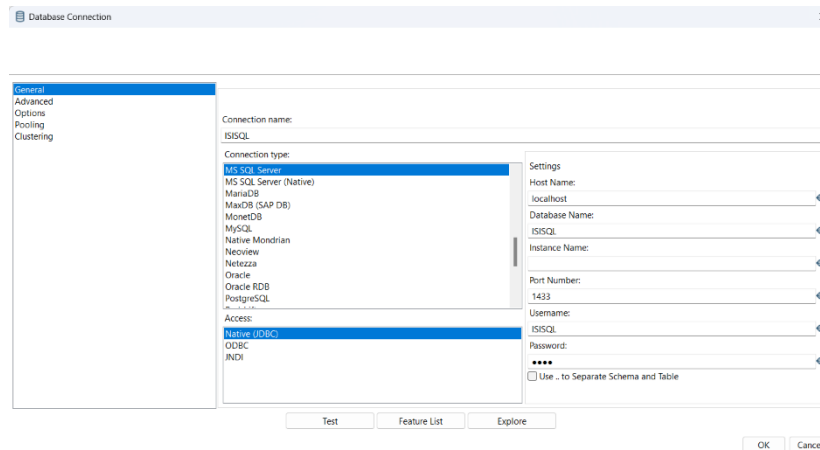


Figura 1.2: Conexão realizada no Pentaho

Após isso, foi criada a tabela no SQL server com a seguinte Query

```
1. CREATE TABLE MusicData (
2. track_name NVARCHAR(255) NOT NULL PRIMARY KEY,
3. artists_name NVARCHAR(255) NULL,
4. artist_count INT NULL,
5. released_year INT NULL,
6. released_month INT NULL,
7. released_day INT NULL,
8. in_spotify_playlists BIT NULL,
9. in_spotify_charts BIT NULL,
10. streams NVARCHAR(255) NULL,
11. in_apple_playlists BIT NULL,
12. in_apple_charts INT NULL,
13. in_deezer_playlists BIT NULL,
14. in_deezer_charts INT NULL,
15. in_shazam_charts NVARCHAR(255) NULL,
16. bpm DECIMAL(5,2) NULL,
17. key NVARCHAR(255) NULL,
18. mode NVARCHAR(50) NULL,
19. danceability_percent DECIMAL(5,2) NULL,
20. valence_percent DECIMAL(5,2) NULL,
21. energy_percent DECIMAL(5,2) NULL,
22. acousticness_percent DECIMAL(5,2) NULL,
23. instrumentalness_percent DECIMAL(5,2) NULL,
24. liveness_percent DECIMAL(5,2) NULL,
25. speechiness_percent DECIMAL(5,2) NULL,
26. cover_url NVARCHAR(500) NULL
27. );
```

Figura 1.3: Query para criação da tabela no SQL



# Transformações

No Pentaho Kettle, as transformações são usadas para realizar processos de ETL, ou seja, extração, transformação e carga de dados. Na fase de extração, os dados podem ser obtidos de várias fontes, como ficheiros CSV, Excel, bases de dados, utilizando passos como o CSV File Input ou o Table Input.

## Transformações Desenvolvidas

Transformação *htmlAsAmbient*

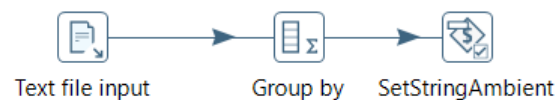


Figura 2: Transformação *htmlAsAmbient*

A figura apresenta a transformação denominada *htmlAsAmbient*.

Inicialmente, o *Text file input*, onde um ficheiro “HTML” é usado para a conversão de dados. Após isso, a operação *Group by* agrupa os dados com base em uma ou mais colunas específicas, permitindo a agregação de informações. Por fim, a etapa *SetStringAmbient* aplica uma transformação nos dados agrupados, provavelmente definindo ou ajustando o valor de uma string relacionada ao ambiente HTML.

Essa transformação permite organizar e manipular os dados provenientes de um ficheiro de texto, ajustando-os de forma adequada para serem utilizados no contexto de HTML, como parte do processo de preparação do conteúdo para ser enviado ou exibido

## Transformação 1

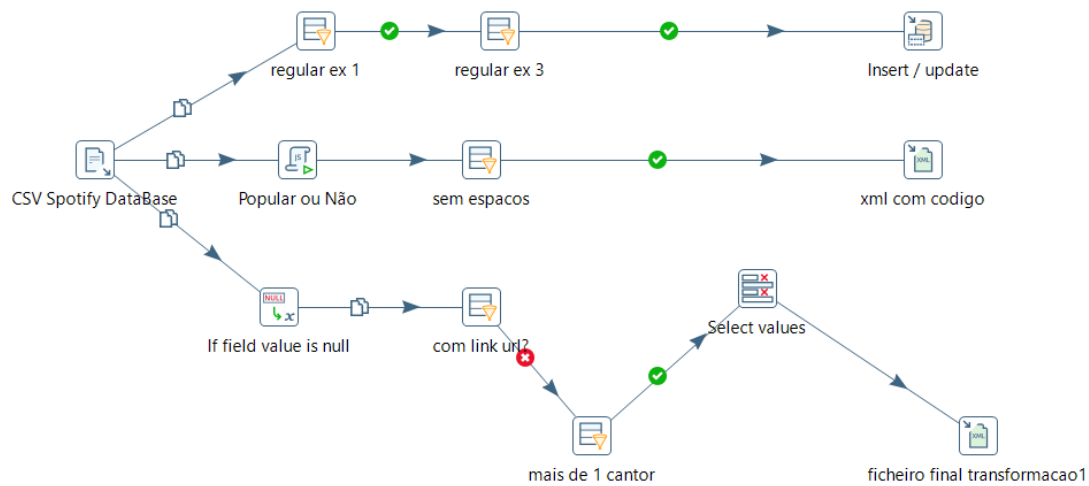


Figura 3: Transformação 1

Na transformação acima apresentada, existe a leitura de informação a partir de um ficheiro CSV (ficheiro inicial que contém a informação inicial).

Nesta transformação, são efetuadas diferentes operações. Na primeira, foram realizados dois Filter Rows, para verificar a existência de caracteres especiais em 2 campos diferentes (track\_name e artist\_name), após essa avaliação, são inseridos na base de dados SQL (Insert/Update).

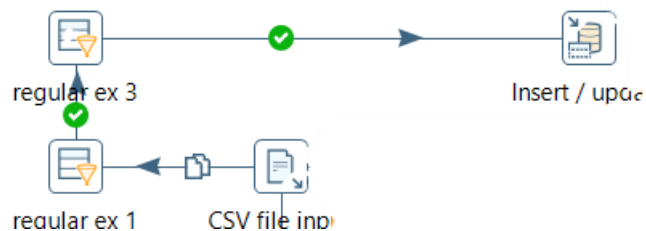


Figura 3.1: Passos de enviar dados para a base de dados

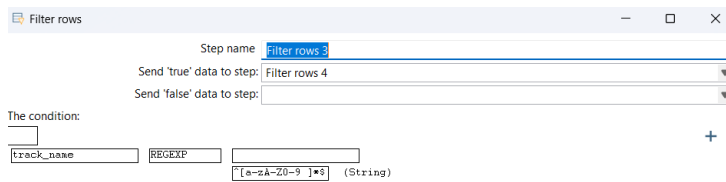


Figura 3.2: Filter Rows onde expressão regular

Aqui, verifica-se se o no campo pretendido, apenas existem caracteres que podem ser lidos, ou seja, sem caracteres especiais.

Além disso, são feitas mais alterações/configurações dos dados para conseguir feita a sua manipulação de forma correta. Um desses exemplos é a utilização do passo “If field value is null”, onde são atribuídos valores assumir se o valor do campo for nulo.

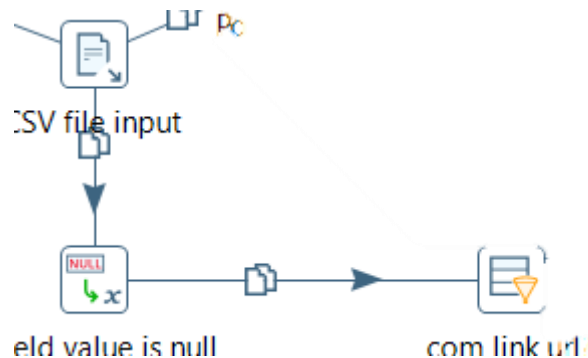


Figura 3.3: If field value is null Transformação 1

Field	Replace by value	Conversion mask (Date)	Set empty string?
1 track_name			N
2 artists_name			N
3 artist_count			N
4 released_year			N
5 released_month			N
6 released_day			N
7 in_spotify_playlists			N
8 in_spotify_charts			N
9 streams			N
10 in_apple_playlists			N
11 in_apple_charts			N
12 in_deezer_playlists			N
13 in_deezer_charts			N
14 in_shazam_charts	-1		N
15 bpm			N
16 key	NA		N
17 mode			N
18 # of non-zero values in...			N

Figura 3.4: If field value is null

Ao vermos esta imagem, podemos ver que no campo *in\_shazam\_charts*, vai ser atribuído o valor -1 e no campo *KEY* o valor NA.

Ao longo desta transformação são realizadas mais algumas filtrações, como por exemplo, se contém “link\_url” ou se a música tem mais de 1 artista (contém “,” no campo).

cover\_url =   
Not Found (String)

Figura 3.5: Verificação se a música tem link\_url

artists\_name CONTAINS   
,(String)

Figura 3.6: Verificação se a música tem mais de 1 artista

Alguns caracteres podem não ser lidos com sucesso, como por exemplo o “%”, com um *Select Value* é possível renomear o campo e assim resolver esse problema.



Figura 3.7: Renomeação dos campos com o Select Values

Após isso, ser tudo gravado com sucesso, o conteúdo desta transformação é armazenado num ficheiro do tipo XML.

## Transformação 2

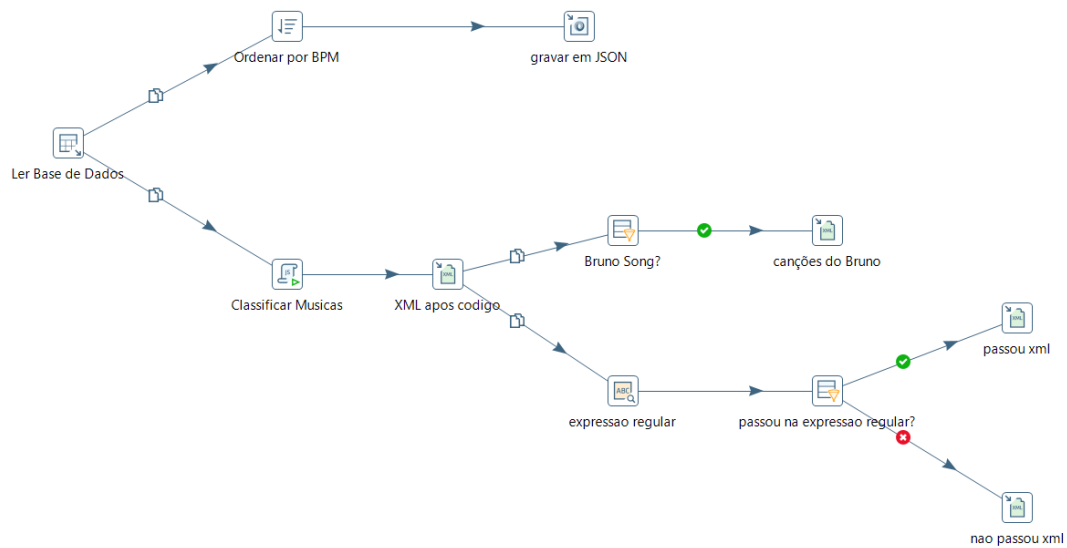


Figura 4: Transformação 2

Esta transformação começa com a leitura da base de dados que foi gravada na Transformação 1.

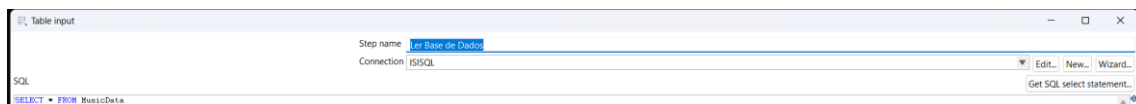


Figura 4.1: Acesso a Base de dados pretendida e query SQL

A partir desses dados lidos, são executadas algumas operações para trabalhar com os dados

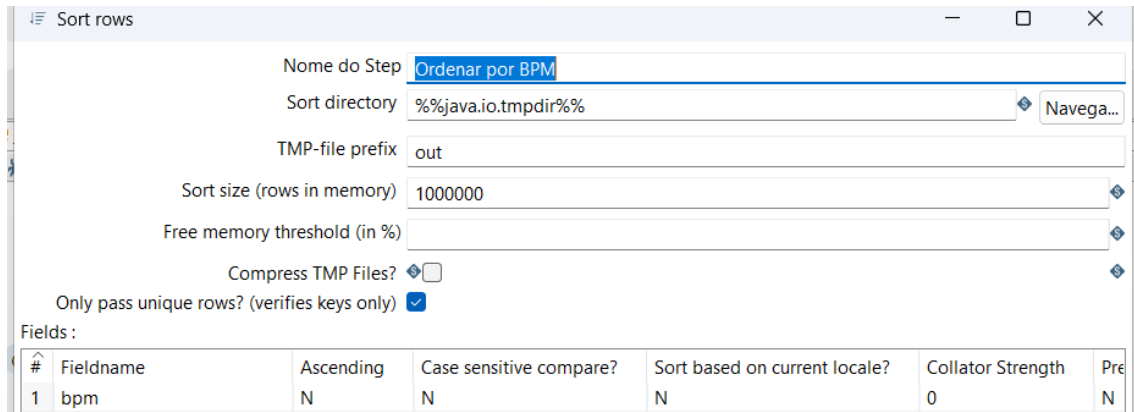


Figura 4.2: Ordenação pelos BPM de forma decrescente

Na imagem acima, recorrendo a um Sort Rows, tendo em conta o seu valor de BPM, as músicas são ordenadas de forma decrescente e armazenadas num ficheiro JSON.

Também foram realizadas operações recorrendo a código JavaScript em que são analisadas e atribuídas novas colunas com as conclusões retiradas a partir do código,

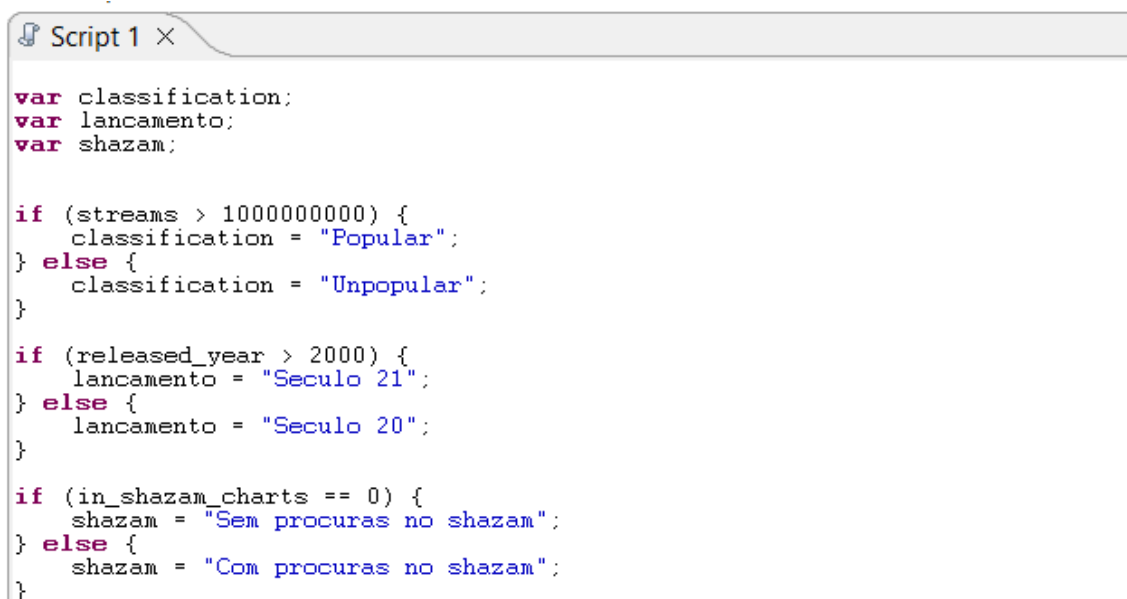


Figura 4.3: Script com o código

A partir deste código, são criadas, para cada música, 3 novas variáveis, classification, lançamento e shazam, tendo em conta as conclusões retiradas a partir do algoritmo e são guardados num ficheiro XML.

Após isso, recorrendo a um Regex Evaluation, verifica-se, se o campo Artists\_name não tem espaços

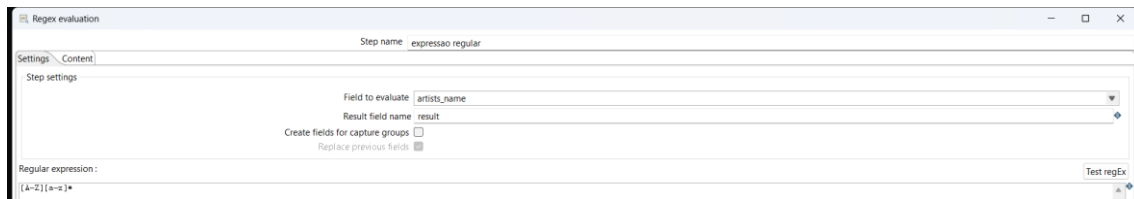


Figura 4.4: Regex Evaluation

O resultado deste Regex Evaluation é uma avaliação, se passou atribui um “Y”, se não passou, atribui um “N” e com isso serão filtrados os que passaram ou não.

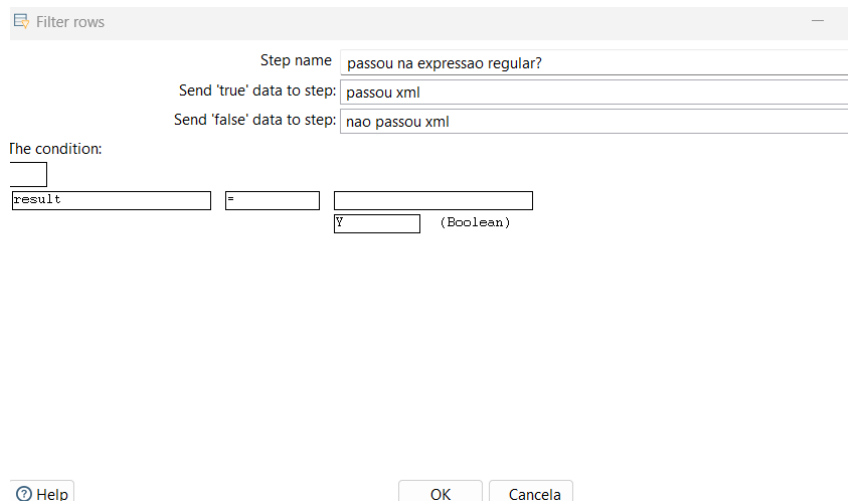


Figura 4.5: Verificação se passou ou não

Aqui analisa-se o resultado da avaliação, separa-se tendo em conta o resultado da avaliação da expressão regular.

Se passou vai para um ficheiro XML e se não passou para outro

Filter rows

Step name: Bruno Song?

Send 'true' data to step: canções do Bruno

Send 'false' data to step:

The condition:

☐ artists\_name CONTAINS Bruno Mars (String)

Help OK Cancela

Figura 4.6: Verificação de Cantor

Aqui verifica-se se no campo Artists\_name contém “Bruno Mars” e caso sim, armazena-se num ficheiro de XML.



## Transformação 3

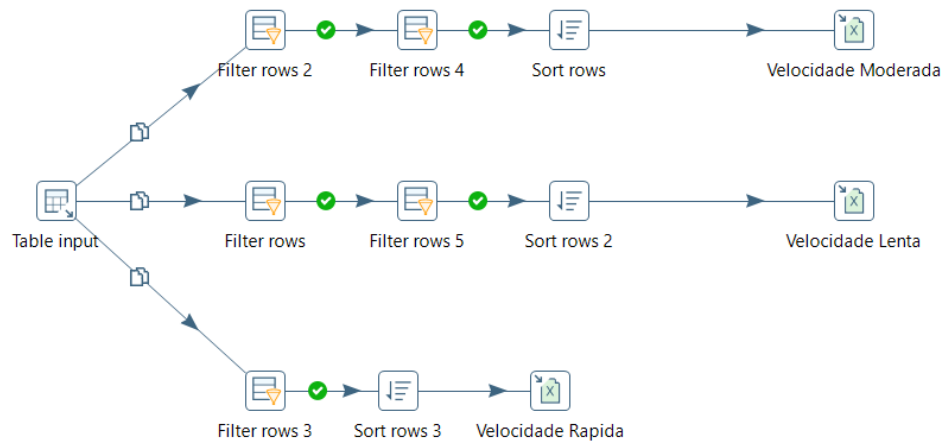


Figura 5: Transformação 3

Esta transformação começa com a leitura da base de dados SQL gravada na transformação 1.

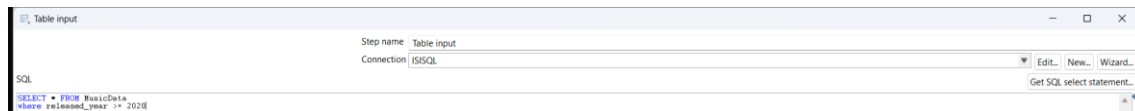


Figura 5.1: Acesso a Base de dados pretendida e query SQL

A partir desses dados lidos, são executadas algumas operações para trabalhar com os dados.

Como se pode ver pela imagem acima, as operações que se realizaram serão apenas com músicas da década dos anos 2020's.

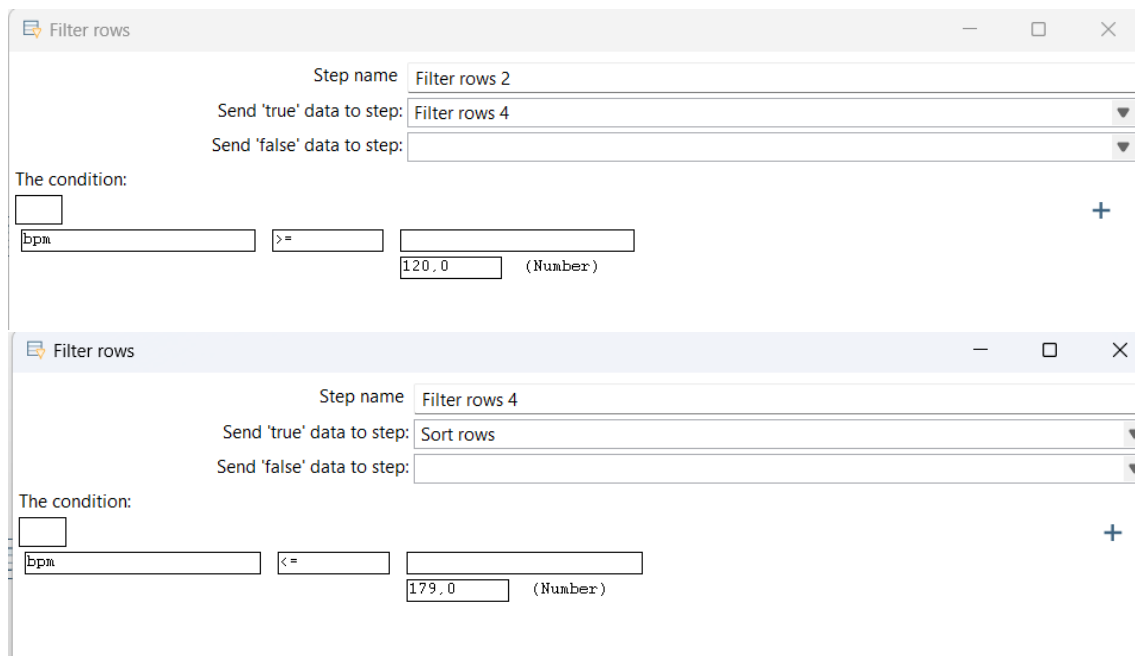
A partir disso, iremos classificar as músicas a partir dos seus BPM's (Batidas por minuto) e serão classificadas como: velocidade lenta, moderada e rápida.

Aqui estão os critérios para a classificação acima:

BPM	Classificação
60>=BPM<=119	Velocidade Lenta
60>=BPM<=179	Velocidade Moderada
180>=BPM	Velocidade Rápida

*Figura 5.2: Critério de Classificação quanto aos BPM's*

Todas estas filtragens foram realizadas a partir dos passos “Filter Rows”, que estão representados na imagem da Transformação 3. Mas segue-se abaixo, como exemplo, as filtragens para as Músicas de Velocidade Moderada.



The image displays two screenshots of the 'Filter rows' configuration window in a software tool, illustrating the setup for filtering music by BPM (Beats Per Minute) for the 'Velocidade Moderada' category.

**Top Screenshot (Filter rows 2):**

- Step name:** Filter rows 2
- Send 'true' data to step:** Filter rows 4
- Send 'false' data to step:** (Empty)
- The condition:** bpm >= 120,0 (Number)

**Bottom Screenshot (Filter rows 4):**

- Step name:** Filter rows 4
- Send 'true' data to step:** Sort rows
- Send 'false' data to step:** (Empty)
- The condition:** bpm <= 179,0 (Number)

*Figura 5.3: Filtragens para as músicas de Velocidade Moderada*

Após esses processos, obtemos os grupos de músicas classificados pelos seus BPM e foram ordenadas por ordem decrescente recorrendo ao passo Sort Row.

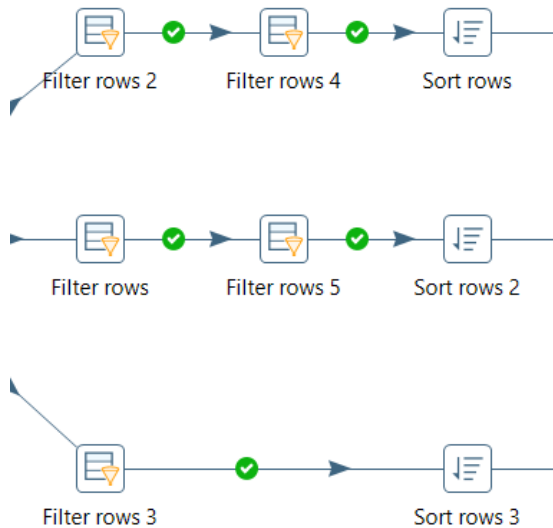


Figura 5.4: Chamada dos passos Sort Rows

Sort rows

Nome do Step: Sort rows

Sort directory: %%java.io.tmpdir%%

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Pre
1	bpm	N	N	N	0	N

Figura 5.5: Filtragem do Sort Rows para ordenar pelos BPM

Após isso, as músicas são guardadas em ficheiros tendo em conta a sua velocidade que foi obtida pelos processos demonstrados anteriormente.

# Jobs

Os jobs permitem incluir várias tarefas, como executar transformações, transferir ficheiros, enviar emails ou executar scripts. Um exemplo de tarefa comum é o passo Start, que define o ponto de início de um job. A partir daí, podem-se configurar passos como o Transformation, que executa uma transformação previamente definida. Um job pode ter condicionais de fluxo, como o Success ou Failure, que permitem tomar decisões com base no resultado da execução de uma tarefa anterior.

## Jobs Desenvolvidos

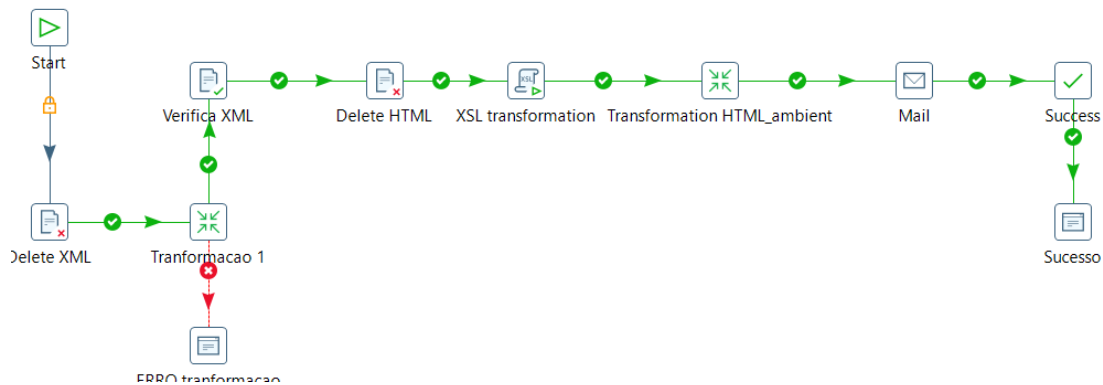


Figura 6: Job 1

Este JOB inicia com um processo de preparação para o tratamento de ficheiros XML e HTML. O primeiro passo é verificar que o ficheiro XML está eliminado.



Figura 6.1: Eliminação do ficheiro XML

Após essa verificação, é realizada uma transformação (a Transformação 1) onde é gerado um Ficheiro XML que vai ser utilizado futuramente.

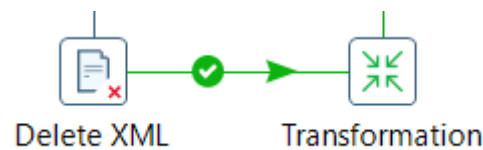


Figura 6.2: Evocação da Transformação

Caso ocorra um erro na transformação, é executado um log de erro

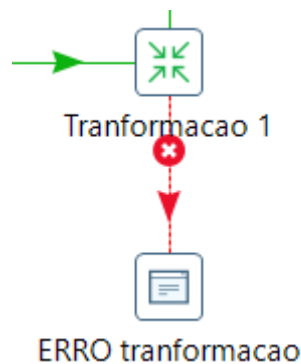


Figura 6.3: Log erro Execução da Transformação 1

Após isso, o mesmo ficheiro XML é verificado para garantir que está no formato correto e que a sua estrutura é válida. Este processo é fundamental para assegurar que as operações subsequentes possam ser realizadas sem problemas. Após a verificação do XML, ocorre uma igual para o ficheiro HTML, eliminando ficheiros antigos que poderiam interferir no novo processamento.



Figura 6.4: Verificações Necessárias

Posteriormente, ocorre a transformação XSL, que converte o ficheiro XML num ficheiro HTML formatado de acordo com uma folha de estilo predefinida. Esta transformação é fundamental para a apresentação correta dos dados extraídos do XML.

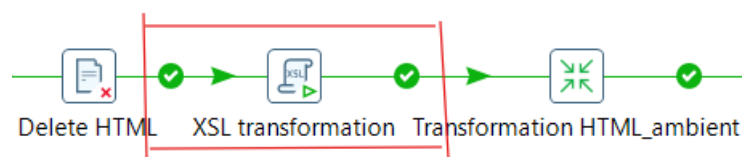


Figura 6.5: XSL Transformation

Após a transformação XSL, é realizada uma nova transformação no ficheiro HTML, denominada "Transformation HTML\_ambient". Esta fase parece ajustar ou refinar o ambiente do ficheiro HTML, possivelmente aplicando formatações adicionais ou realizando ajustes nos dados antes do ficheiro ser enviado.

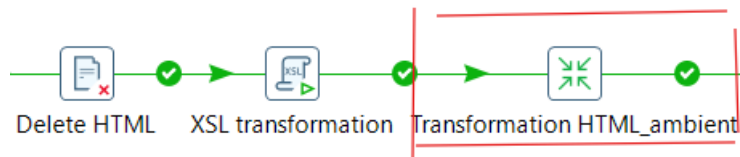


Figura 6.6: Transformation HTML\_ambient

Com o HTML preparado, o próximo passo é o envio de um e-mail, que pode conter o ficheiro HTML gerado ou simplesmente uma notificação sobre o sucesso ou falha do processo. Após o envio do e-mail, o JOB é concluído com sucesso, registrando que todo o processo foi finalizado corretamente.



Figura 6.7: Envio de Email

Após todo o processo, é impresso na tela um log de sucesso.

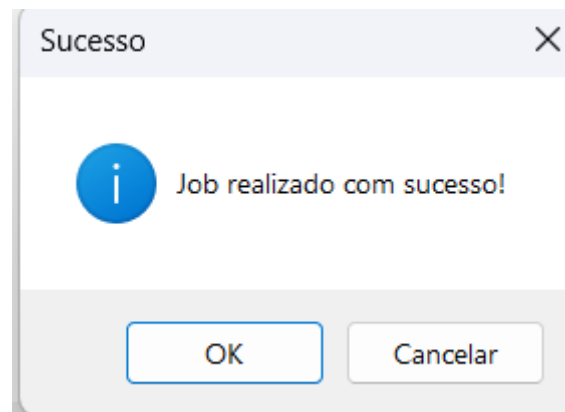


Figura 6.8: Log de Sucesso!

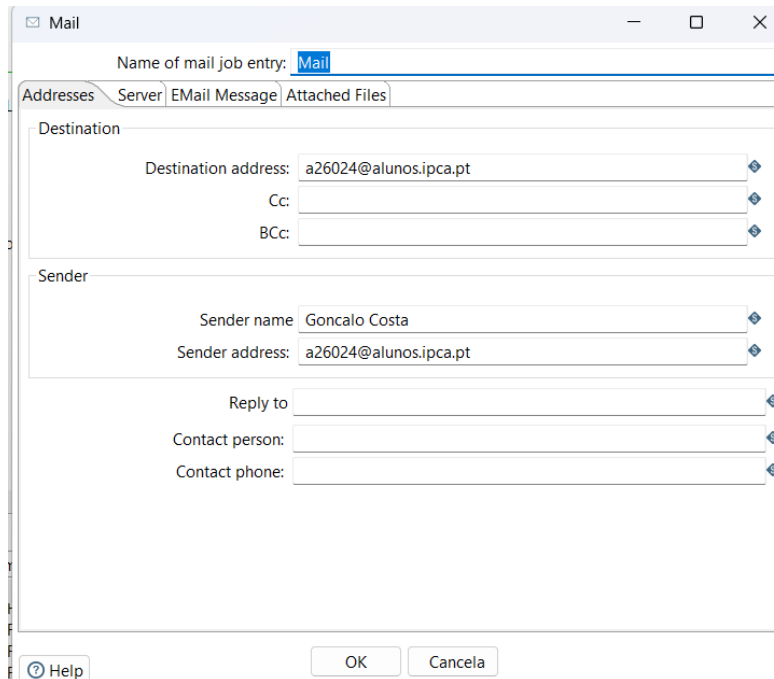
# Email



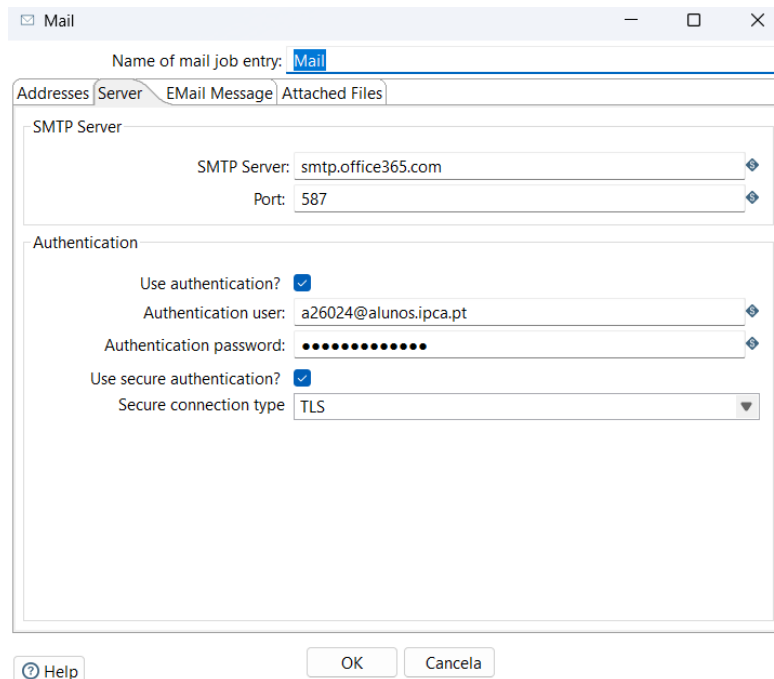
*Figura 7: Email que chegou à caixa de entrada do destinatário*

Como pode ser visto pela imagem acima, o email chegou ao endereço de destino com sucesso.

Isto deve-se à sua correta configuração

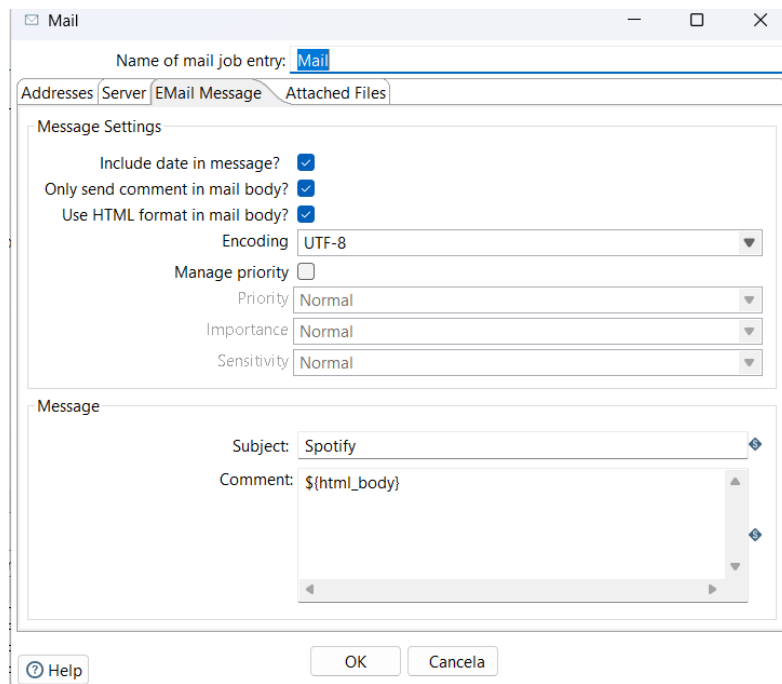


*Figura 7.1: Configuração de Email (destination e sender address)*



*Figura 7.2: Configuração de Email (server)*





*Figura 7.3: Configuração de Email (mensagem de email)*

# Node-RED

O Node-RED é uma ferramenta de desenvolvimento baseada em fluxos que permite conectar dispositivos, APIs e serviços online de maneira fácil e visual. Ele foi originalmente desenvolvido pela IBM e é amplamente utilizado para Internet das Coisas (IoT), automação e integração de sistemas.

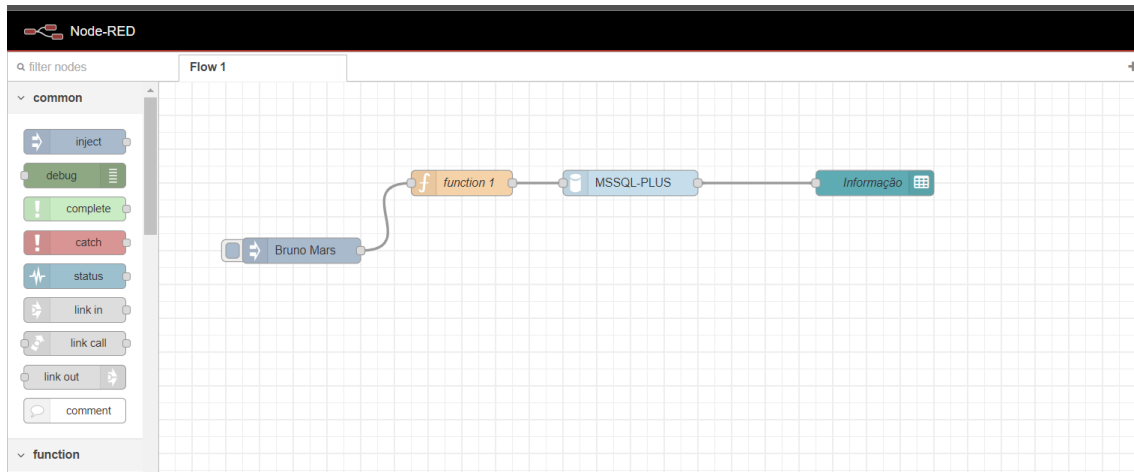


Figura 8: Esquema realizado no Node-RED

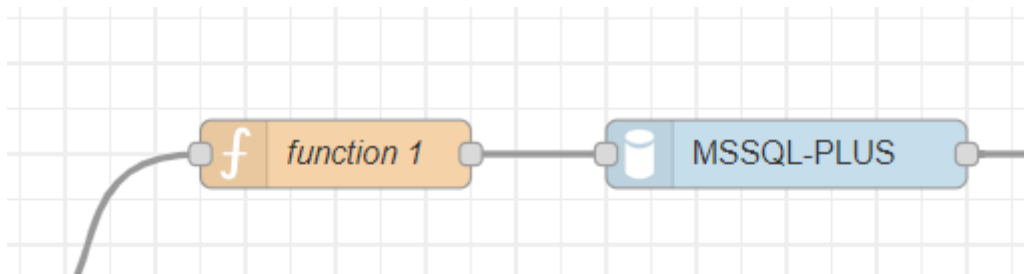


Figura 8.1: Chamada da function e conexão com SQL

Aqui, é chamada uma function com uma query para procurar na Base de Dados SQL informações pretendidas pelo utilizador.

A conexão com o SQL é feita pelo MSSQL-PLUS apresentado na imagem

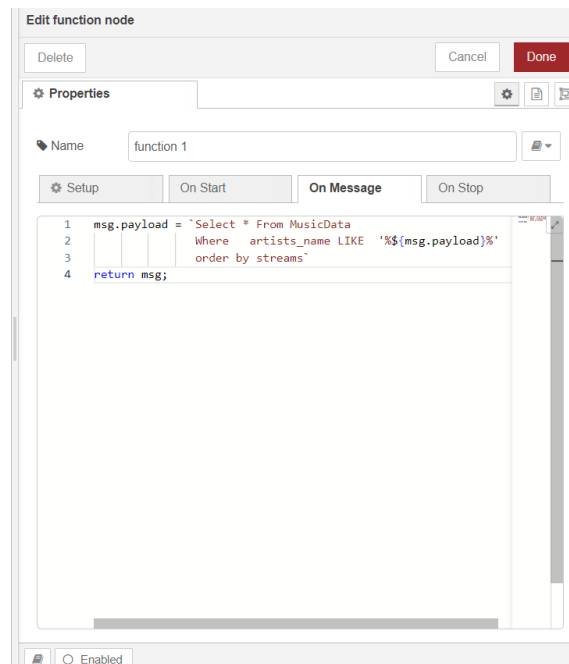


Figura 8.2: Query SQL

Músicas de Cantor																			
Tabela de informação																			
track_name	artists_name	artist...	r...	r...	r...	i...	i...	s...	i...	i...	i...	i...	b...	key	m...	d...	v...	e...	a...
Talking To The Moon	Bruno Mars	1	2010	10	4	true	true	10629...	5	0	862	0	0	146	C#	Minor	52	7	61
Locked Out Of Heav...	Bruno Mars	1	2012	12	5	true	true	14813...	0	0	356	0	0	144	F	Major	73	87	70
Just The Way You Are	Bruno Mars	1	2010	1	1	true	true	16414...	82	0	2946	0	0	109	F	Major	63	46	85
When I Was Your Man	Bruno Mars	1	2012	12	5	true	true	16611...	0	0	806	0	0	145		Major	60	43	27

Figura 8.3: Resultado do Esquema de Node-RED

Aqui, está apresentado o resultado de todo o processo apresentado anteriormente

## Vídeo demonstrativo



*Figura 9: QR Code*

Ao entrar neste QR Code, será redirecionado para um perfil, escolha a opção vídeo para o ver.

## Conclusão

Neste contexto, conclui-se que o trabalho permitiu consolidar os conhecimentos relacionados com os processos de ETL, desde a extração, transformação e carga de dados, até à aplicação prática de ferramentas como o Pentaho Kettle. A integração de ferramentas complementares, como o Node-RED e o Home Assistant, foi uma oportunidade para explorar novas tecnologias e potenciar a automação de processos.

Os desafios enfrentados, como o tratamento de grandes volumes de dados e a manipulação de diferentes formatos (XML, JSON, bases de dados), foram superados através da utilização de técnicas adequadas, como o uso de expressões regulares, normalização de dados. A utilização de processos de logging e visualização dos resultados também se revelou uma mais-valia para monitorizar e validar a eficiência dos processos desenvolvidos.

Em suma, este trabalho proporcionou uma experiência prática significativa, permitindo a exploração de tecnologias atuais no contexto da Integração de Sistemas de Informação, reforçando a capacidade de desenvolver soluções escaláveis e eficientes.

## **Bibliografia/Web grafia**

- Material disponibilizado pelo professor da unidade curricular
- Sebenta Pentaho Kettle
- Node-Red Cook-Book
- SQL Server Sebenta