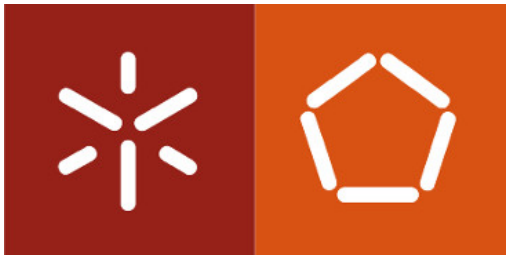

Universidade do Minho

Mestrado em Engenharia Informática

Trabalho Prático

Carolina Resende Marques, PG42818

Francisco Borges, PG42829

Rui Pereira, PG42853

Vasco António Lopes Ramos, PG42852

Análise de Dados
4º Ano, 1º Semestre
Departmento de Informática

28 de abril de 2021

Índice

1 Introdução **1**

2 Análise e Escolha da Fonte de Dados e Ferramentas Utilizadas **2**

2.1 Fonte de Dados (*Dataset*) 2

2.2 Ferramentas Utilizadas 5

2.2.1 Processo ETL 5

2.2.2 Desenvolvimento do *Data Warehouse* 6

2.2.3 *Business Intelligence* 6

3 Arquitetura do *Data Warehouse* **8**

3.1 Método dos 4 Passos para a Modelação Dimensional 8

3.1.1 Seleção da Área de Suporte à Decisão (Processo de Negócio) 8

3.1.2 Definição da granularidade dos dados 9

3.1.3 Seleção das Dimensões de Análise 9

3.1.4 Identificação dos factos 10

3.2 Modelo Dimensional 10

4 Processos de *ETL* **12**

4.1 Extração 12

4.2 Transformação 13

4.2.1 Classifcação em Gamas de Valores 13

4.2.2 Separação dos valores das listas 14

4.3 Carregamento 15

5 Sistema de *Business Intelligence* **16**

5.1 Artistas com maior número de músicas 17

5.2 Anos mais Populares 18

5.3 Artistas mais Populares 19

5.4 Número de Artistas Populares por Década 19

5.5 1929 vs 2020 20

5.6 Artistas Pop Populares por Duração 22

5.7 Géneros de Música Mais Populares 23

5.8 Rácio entre Qualidade de Som e Popularidade 24
5.9 Rácio entre energia e Nível Acústico 24
5.10 Relação entre a Emoção e a Capacidade de Dançar 25
5.11 Músicas Rock Populares e a Sua Energia 26

6 Conclusão **28**

A Carregamento dos dados no *Data Warehouse* **29**

Referências **32**

Lista de Figuras

1 Arquitetura *Data Warehouse & Business Intelligence* (fonte: [5]) . . 8
2 Dimensões Selecionadas 9
3 Factos Selecionados 10
4 Modelo Dimensional 11
5 Número de músicas por artista 17
6 Anos Mais Populares 18
7 Artistas mais populares 19
8 Número Artistas Populares por Década 20
9 Valence na queda da bolsa (1929) e valence em 2020 (pandemia) . . 21
10 Artistas Pop populares por duração 22
11 Top 3 Géneros mais Populares 23
12 Rácio entre Qualidade de Som e a Popularidade 24
13 Rácio entre energia e accousticness 25
14 Relação entre Emoção e a sua Capacidade de Dançar 26
15 Relação entre a popularidade de uma música Rock e a sua energia . 27

Lista de Códigos

1 Função de classficação do tempo de duração de uma música 13
2 Função de classficação da proprieadade *valence* 14
3 *Procedure* para separar os valores presentes nas listas de géneros . . 14
4 Carregamento dos dados no *Data Warehouse* 29

1 Introdução

Na atualidade, o tratamento, análise e exploração de dados é uma das principais metodologias que as empresas dispõem como suporte ao seu processo de tomada de decisões e um fator essencial na otimização dos seus processos internos. Assim, os agentes de decisão procuram por ferramentas que permitam manipular e visualizar os dados segundo as várias vistas/perspetivas do seu negócio.

Empresas como *Facebook*, *Google* e *Spotify* demonstraram que em todos os segmentos de mercado a utilização de tecnologias analíticas são uma grande vantagem competitiva.

Para este fim, destacam-se as ferramentas de *ETL*, *Business Intelligence* e *Business Analytics* que permitem automatizar e otimizar os processos de análise de negócio através dos dados produzidos pelo mesmo. Algumas destas ferramentas são analisadas e distrbuídas consoante o seu propósito e aceitação no mercardo pela empresa Gartner, disponível em [1].

Este projeto adotou a perspetiva de uma equipa de IT numa empresa dentro da indústria musical. A empresa pretende potenciar o seu processo de tomada de decisão para perceber quais as tendências do mercado. Com a ajuda de uma ferramenta de BI quer identificar quais os estilos musicais em que devem apostar, quais as características que os seus *singles* deverão ter e também como a Indústria tem evoluído com o tempo.

Ao longo deste documento iremos apresentar o *dataset* do *Spotify* utilizado pela empresa para obter informação, a arquitetura do sistema de *Data Warehouse*, os processos de ETL implementados para consolidar informação, a ferramenta de BI para consulta de informação e, por fim, as respetivas conclusões.

Deste modo, o objetivo deste projeto inclui estudar e aplicar técnicas de *Business Intelligence* e *Business Analytics*, utilizando como fonte de dados o histórico de músicas disponibilizadas e distribuídas pelo *Spotify*.

2 Análise e Escolha da Fonte de Dados e Ferramentas Utilizadas

2.1 Fonte de Dados (*Dataset*)

Como *dataset* para este trabalho, como já referido acima, escolheu-se uma coletânea de informações relacionadas com as músicas, artistas e géneros distrbuídos na plataforma *Spotify* desde o ano de 1921 até 2020, disponível em [2]. Parte da dificuldade associada à procura e escolha do *dataset* teve que ver com a necessidade de encontrar um *dataset* que fosse grande o suficiente (em número de registos) para servir o propósito do trabalho e que tivesse informação relevante e útil que permitisse construir uma análise interessante e de valor acrescentado.

A principal razão da escolha deste *dataset* relaciona-se com o facto da indústria musical ter evoluído com a sociedade ao longo dos anos, pelo que a inovação no som reflete a evolução cultural e tecnológica, bem como pelo interesse em perceber que tipo de características são, de um modo geral, mais importantes nas músicas de forma a torná-las populares, recordáveis ou importantes no espetro da sociedade mundial.

Relativamente ao *dataset* em si, este conta com 5 ficheiros CSV diferentes

- ***data.csv*** - contém uma listagem de todas as músicas, os artistas associados e as métricas/campos associados às músicas.
- ***data_by_artist.csv*** - contém todos os artistas e as respetivas métricas/-valores associados a estes.
- ***data_by_genres.csv*** - contém todos os géneros musicais e as respetivas métricas/valores associados a estes.
- ***data_by_year.csv*** - contém todos os anos desde 1921 até 2020 e as respetivas métricas/valores associados a cada um dos anos.
- ***data_w_genres.csv*** - contém todos os artistas, bem como os géneros associados a cada um e as respetivas métricas/valores associados aos artistas.

Deste modo, tendo em conta que o ***data_by_artist.csv*** e ***data_by_genres.csv*** estão perfeitamente representados no ***data_w_genres.csv*** e que as métricas agrupadas por ano conseguem ser obtidas através de uma correta modelação dos dados (descartando, então, o ***data_by_year.csv***), os ficheiros CSV que foram efetivamente utilizados para a construção do *data warehouse* foram: ***data.csv*** e ***data_w_genres.csv***.

Relativamente às métricas/valores representados em cada um destes ficheiros CSV são os seguintes.

No ficheiro CSV ***data.csv*** existem os seguintes campos:

- *id* - identificador único associado a cada música no sistema do *Spotify*;
- *name* - nome da música;
- *artists* - lista de artistas associados à música;
- *year* - ano em que a música foi lançada;
- *release_date* - data em que a música foi lançada (pode aparecer nos formatos yyyy, yyyy-mm ou yyyy-mm-dd);
- *duration_ms* - duração da música, em milisegundos;
- *explicit* - valor binário a identificar se a música tem conteúdo de caráter explicito;
- *mode* - valor binário a identificar se a música começa com uma sequência de acordes;
- *tempo* - o ritmo da música, em número de batimentos por minuto (BPM);
- *valence* - valor contínuo, de 0 a 1, que representa a positividade transmitida pela música;
- *acousticness* - valor contínuo, de 0 a 1, que representa o quão acústica é a música;
- *danceability* - valor contínuo, de 0 a 1, que representa o quão adequada/fácil a música é para dançar;

- *energy* - valor contínuo, de 0 a 1, que representa a intensidade/energia da música;
- *instrumentalness* - valor contínuo, de 0 a 1, que representa a instrumentalidade da música (se tem mais parte intrumental ou vocal);
- *key* - o valor da nota musical (chave) mais predominante na música;
- *liveness* - valor contínuo, de 0 a 1, que representa a duração relativa da música que tem componente de atuações ao vivo;
- *loudness* - valor contínuo, de 0 a 1, que representa a qualidade do som da música;
- *popularity* - a popularidade da música, recentemente, nos Estados Unidos da América;
- *speechiness* - valor contínuo, de 0 a 1, que representa a presença da voz (humana) na música.

No ficheiro CSV ***data_w_genres.csv*** existem os seguintes campos (os que já foram definidos acima não serão novamente definidos, sendo que representam o mesmo conceito, mas associado ao artista):

- *artist* - nome do artista;
- *genres* - lista de géneros associados ao artista;
- *duration_ms*;
- *valence*;
- *acousticness*;
- *danceability*;
- *energy*;
- *instrumentalness*;
- *key*;
- *liveness*;

- *loudness*;
- *popularity*;
- *speechiness*;
- *count* - o número de músicas associadas ao artista.

De uma forma geral, todos estes campos foram utilzados, à exceção do campo *count* no CSV ***data_w_genres.csv***, já que este pode ser obtido pela modelação do *data warehouse*.

2.2 Ferramentas Utilizadas

2.2.1 Processo ETL

O processo de Extração, Transformação e Carregamento involve diferentes fontes de informação, um constante aumento do volume de dados e diferentes níveis de estruturação de dados. No entanto, tendo em conta o nível de complexidade deste problema em específico, o grupo decidiu que não exigia uma plataforma dedicada.

Para este efeito, a linguagem SQL foi usada como a ferramenta para este processo. SQL (*Structured Query Language*) é uma linguagem declarativa de programação para lidar com bases de dados relacionais (baseadas em tabelas).

Esta linguagem é:

1. **Fácil de usar** - intuitiva e rapidamente se aprende;
2. **Simples** - de fácil compreensão;
3. **Rápido** - tem bom desempenho em termos de custos;

Para além das vantagens enumeradas, membros do grupo já estavam familiarizados com SQL e, por decisão unânime, esta foi escolhida para desenvolver os diferentes scripts usados no processo de ETL.

2.2.2 Desenvolvimento do *Data Warehouse*

Esta parte do trabalho foi realizada através do*MySQL Workbench*, que oferece:

1. Modelação de dados;
2. Desenvolvimento de SQL;
3. Ferramentas de administração abrangente para configuração de servidores;
4. Administração de utilizadores;
5. Backup;
6. etc.

Para além disso a *interface* do utilizador é muito fácil de utilizar, intuitiva e existe a possibilidade de formar e optimizar esquemas e consultas (utilizando ferramentas de visualização gráfica).

Para implementar os modelos resultantes da modelação dimensional houve a necessidade de ter uma ferramenta capaz de reproduzir um modelo lógico e converter este para um modelo físico.

Sendo que foi introduzida aquando o início da unidade curricular de Análise de Dados, o grupo ficou familiarizado com esta ferramenta e, tal como em SQL, houve uma concordância em usá-la e desenvolver, de forma mais rápida e eficaz, o nosso DW.

2.2.3 *Business Intelligence*

Para desenvolver o sistema de *Business Intelligence*, o grupo ponderou se devia ser usada a plataforma *Microsoft Power BI Desktop* [3] ou *Tableau Desktop* [4].

Com base em [1], notou-se que *Tableau* é um Líder no Quadrante Mágico. É afirmado que este oferece uma experiência de exploração de base visual que permite aos utilizadores empresariais aceder, preparar, analisar e apresentar os resultados nos seus dados. Acima deste, também segundo [1], o *Power BI* oferece

preparação de dados, descoberta de dados de base visual, painéis interactivos e análises aumentadas.

A vantagem que levou o grupo a usar esta ferramenta foi a facilidade de exploração visual e manipulação de dados. O *Tableau* permite aos utilizadores ingerir dados rapidamente a partir de uma vasta gama de fontes de dados, misturá-los, e visualizar resultados utilizando as melhores práticas de percepção visual. Os dados podem ser facilmente manipulados durante a visualização, tais como quando se criam grupos, caixas e hierarquias.

Apesar do *Power BI* estar mais bem posicionado segundo a Gartner, como o grupo teve a oportunidade de usar ambas as plataformas foi discutido que houve realmente uma maior facilidade de compreensão ao usar o *Tableau*, pelo que a escolha final foi a utilização do *Tableau*.

3 Arquitetura do *Data Warehouse*

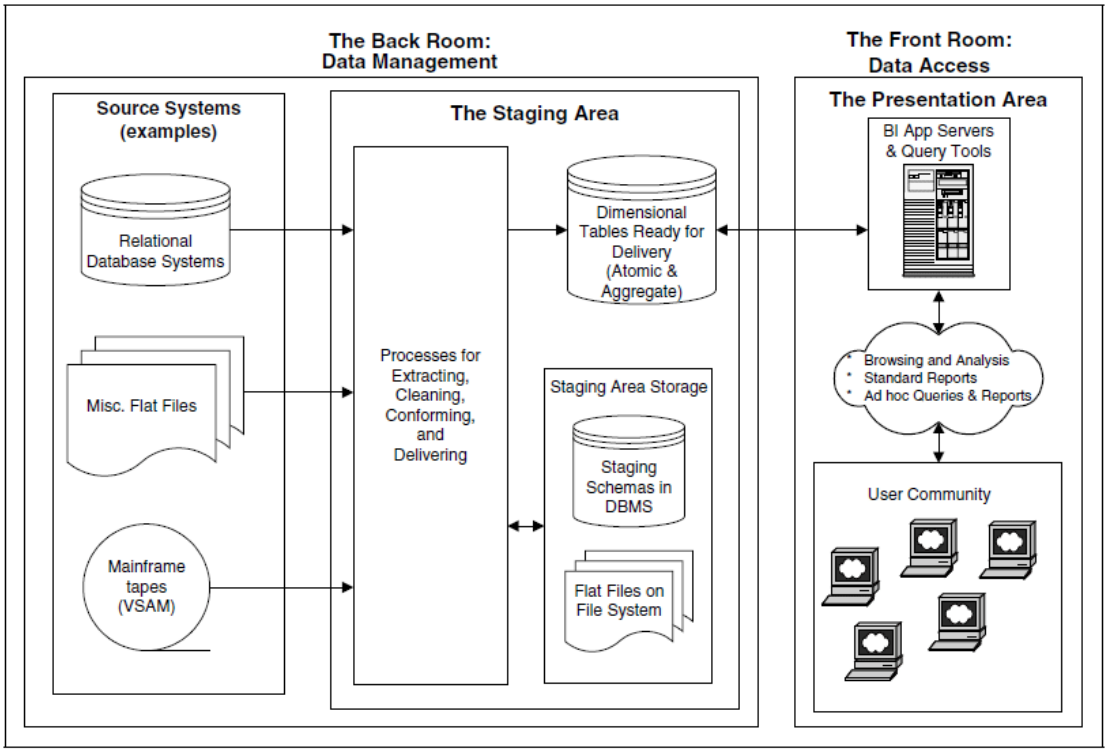

Figura 1: Arquitetura *Data Warehouse* & *Business Intelligence* (fonte: [5])

3.1 Método dos 4 Passos para a Modelação Dimensional

3.1.1 Seleção da Área de Suporte à Decisão (Processo de Negócio)

A construção de um *Data Warehouse* deve ter como foco inicial decidir o seu propósito, isto é, qual a área de suporte à decisão que este irá servir.

Neste caso, parte-se do objetivo (como se fosse uma editora/produtora musical) de perceber quais os fatores que influenciam o mercado da música, tendo em conta as caracterísitcas inerentes às músicas e a evolulção da sociedade (tendo em conta o fator tempo). Desta forma, o processo de negócio a modelar terá principal foco nas **músicas**, **artistas** e **géneros**.

Isto permite que se analise quais os artistas com mais músicas, a qual a relação entre a popularidade de uma música e as suas características, a evolução

da popularidade a nível de artistas ao longo das décadas, entre outros fatores de importância.

3.1.2 Definição da granularidade dos dados

Tendo em conta que neste contexto podia ser vantajoso, de um ponto de vista mais "micro"analisar fatores de músicas/artistas específicos para uma análise de mercado mais localizada, achou-se por bem utilizar os dados ao nível mais atómico possível de forma a, como foi dito, dar maior flexibilidade analítica e permitir que os dados sejam limitados ou agrupados livremente no consumo dos mesmo (nas consultas analíticas).

3.1.3 Seleção das Dimensões de Análise

Uma vez definida a granularidade desejada, torna-se relativamente simples selecionar quais as dimensões adequadas, sendo que neste caso há dimensões em "cascata". Para melhor se perceber essas relações serão demonstradas as dimensões e quais se relacionam, tal como se pode ver na figura 2.

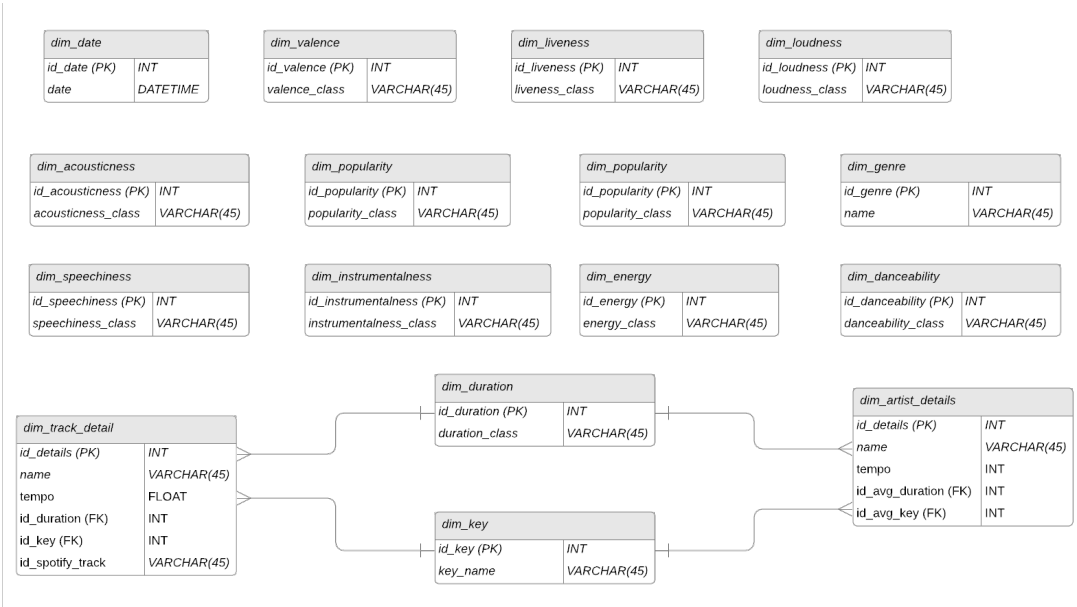

Figura 2: Dimensões Selecionadas

Como se pode ver, nas dimensões que representam métricas (*speechiness*,

popularity, etc), decidiu-se classificar os seus valores por gamas de valores, sendo que o que essas dimensões representam são precisamente essas gamas de valores.

3.1.4 Identificação dos factos

Como último passo, impõe-se a identificação dos factos. No contexto deste trabalho, foram selecionadas duas tabelas de factos: uma para as **músicas** e outra para os **artistas**, como se pode ver na figura 3.

fact_track	
id_track (PK)	INT
mode	TINYINT
explicit	TINYINT
id_release_date (FK)	INT
id_valence (FK)	INT
id_details (FK)	INT
id_acousticness (FK)	INT
id_danceability (FK)	INT
id_energy (FK)	INT
id_instrumentalness (FK)	INT
id_liveness (FK)	INT
id_loudness (FK)	INT
id_popularity (FK)	INT
id_speechiness (FK)	INT

fact_artist	
id_artist (PK)	INT
mode	TINYINT
id_valence (FK)	INT
id_details (FK)	INT
id_acousticness (FK)	INT
id_danceability (FK)	INT
id_energy (FK)	INT
id_instrumentalness (FK)	INT
id_liveness (FK)	INT
id_loudness (FK)	INT
id_popularity (FK)	INT
id_speechiness (FK)	INT

Figura 3: Factos Selecionados

3.2 Modelo Dimensional

Esta secção apresenta o modelo que resultou do processo de modelação anterior.

A figura 4 representa o resultado do processo de modelação dimensional. Deste processo resultou um esquema de constelação de factos, sendo que as duas tabelas de facto, como dito em 3 são: ***fact_track*** e ***fact_artist***. Estas duas tabelas de facto estão também ligadas entre si por uma relação *Many-To-Many*, para ser possível ter precisamente a relação entre um artista e as suas músicas.

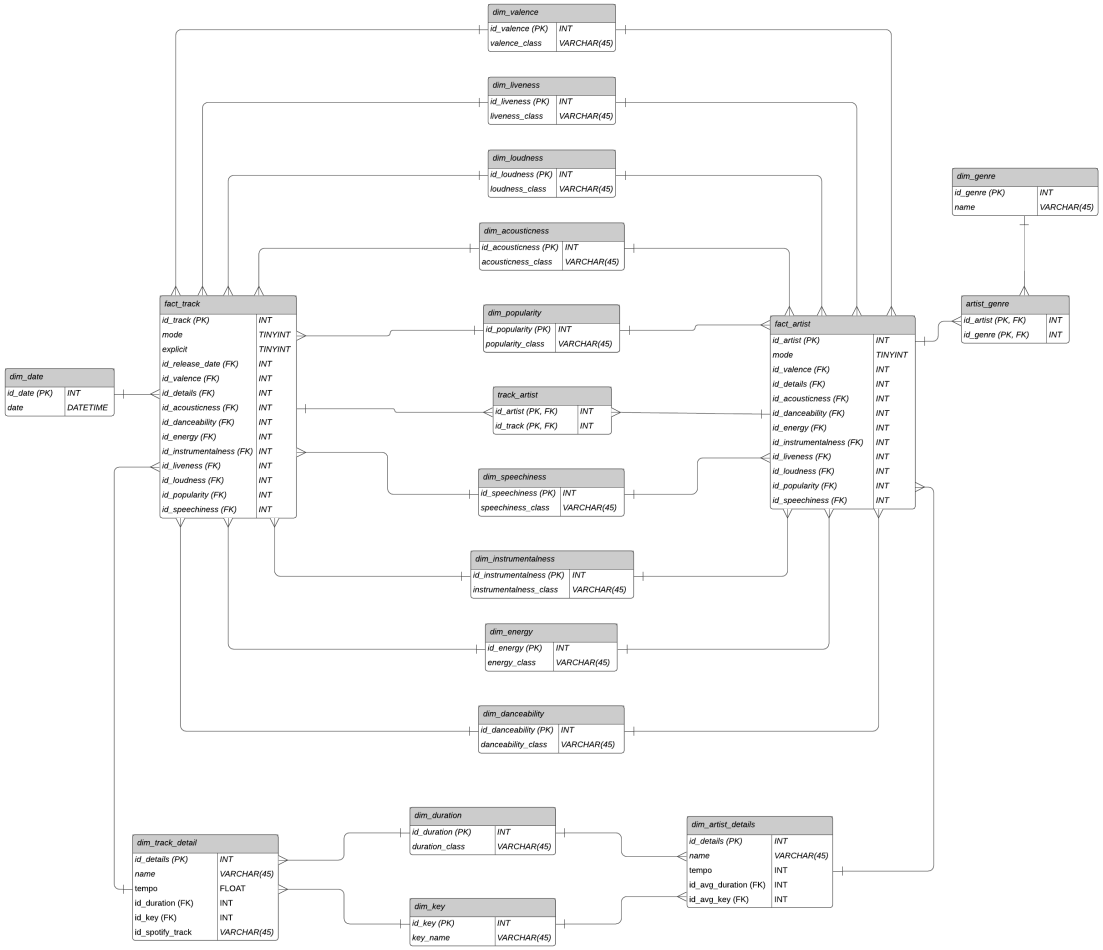

Figura 4: Modelo Dimensional

Uma vez concluído o esquema, aplicou-se o mesmo numa instância de MySQL onde ficará instalado o *Data Warehouse*. Para isto utilizou-se um *script* SQL que serve para criar tanto o *schema* como todas as tabelas e relações.

4 Processos de *ETL*

Não tendo utilizado uma ferramenta específica para os processos de *ETL*, utilizou-se essencialmente a linguagem SQL para o efeito. Para facilitar o processo e cumprir com o que seria expectável, tal como se pode ver na figura 1, criou-se uma *Staging Area* onde os dados são colocados antes da fase final de Load de todo este processo.

Em jeito de nota para o restante desenvolvimento deste capítulo, por se ter usado SQL como ferramenta de ETL existem algumas ações que apesar de estarem listadas como extração também executam alguma parte da transformação e o mesmo acontece na fase da transformação em que também executam algumas operações de população, tal como será possível de perceber de seguida.

4.1 Extração

A primeira fase do processo, o *extract* dos dados, foi feita com recurso ao *Import Wizard* através de um CSV, disponibilizado pelo *MySQL Workbench*, em que os dados são carregados do CSV diretamente para uma tabela com toda a informação. Neste caso isso foi feito para 3 ficheiros CSV: um com os dados das músicas, outro com os dados dos artistas e, por fim, outro com o mapeamento dos *encondings* das notas musicais (necessário para a tabela *dim_key*). Estes *imports* foram feitos para um *schema* diferente do *schema* do *Data Warehouse*. Este novo *schema* corresponde à nossa *Staging Area*.

Durante a execução desta fase deparámo-nos com múltiplos problemas no *import* dos dados, devidos a: problemas de formatação das listas de artistas e géneros, inconsistências na representação de valores numéricos (alguns estavam na representação decimal e outros estavam em notação científica), valores nulos e/ou indefinidos, entre outros. Estes problemas foram resolvidos diretamente nos ficheiros CSV, visto que eram estes erros que impossibilitavam as operações de *import* dos dados. A resolução destes problemas foram resolvidas com *find/replace* e *format* através de expressões *regex*.

Após resolver todos estes problemas e as operações de *import* funcionarem

corretamente, procedeu-se a fazer um *dump* da estrutrura e dados da nossa *Staging Area* para um *script* SQL de forma a facilitar e acelerar o desenvolvimento do restante projeto.

4.2 Transformação

Como referido na secção anterior, algumas transformações tiverem de ser feitas à cabeça nos ficheiros CSV. As restantes, já através de SQL, relacionam-se com operações de agregação de dados na classifcação dos valores das métricas consoante os seus valores (estas transformações são executadas aquando do carregamento final, como será visto na próxima secção) e separação dos valores nas listas dos artistas e géneros.

4.2.1 Classifcação em Gamas de Valores

Para proceder à classficação das métricas em gamas de valores criou-se funções especifícas para cada uma das métricas, onde, através de uma estrutura *switch/ case*, se decide em que gama os valores se enquadram. Os trechos de código 1 e 2 exemplifcam o tipo de funções que foram criadas.

```
DELIMITER |
CREATE FUNCTION duration_classification (duration double)
RETURNS int
DETERMINISTIC
BEGIN
    DECLARE classification int;
    CASE
        WHEN duration <= 60000 THEN SET classification = 1;
        WHEN duration > 60000 AND duration <= 120000 THEN SET classification = 2;
        WHEN duration > 120000 AND duration <= 180000 THEN SET classification = 3;
        WHEN duration > 180000 AND duration <= 240000 THEN SET classification = 4;
        WHEN duration > 240000 AND duration <= 300000 THEN SET classification = 5;
        WHEN duration > 300000 AND duration <= 360000 THEN SET classification = 6;
        ELSE SET classification = 7;
    END CASE;
    RETURN (classification);
END;
|
DELIMITER ;
```

Código 1: Função de classficação do tempo de duração de uma música

```
DELIMITER |
CREATE FUNCTION valence_classification (valence double)
RETURNS int
DETERMINISTIC
BEGIN
    DECLARE classification int;
    CASE
            WHEN valence >= 0 AND valence <= 0.2 THEN SET classification = 1;
            WHEN valence > 0.2 AND valence <= 0.4 THEN SET classification = 2;
            WHEN valence > 0.4 AND valence <= 0.6 THEN SET classification = 3;
            WHEN valence > 0.6 AND valence <= 0.8 THEN SET classification = 4;
        WHEN valence > 0.8 AND valence <= 1 THEN SET classification = 5;
    END CASE;
    RETURN (classification);
END;
|
DELIMITER ;
```

Código 2: Função de classficação da proprieadade *valence*

4.2.2 Separação dos valores das listas

Para tratar da separação dos vários valores presentes nas listas de artistas e géneros foram criados *procedures* de forma a dividir estes valores e populares as respetivas dimensões com esses valores. O trecho de código 3 exemplifica o tipo de *procedures* criados para o efeito.

```
DELIMITER |
CREATE PROCEDURE populate_dim_genre (bound VARCHAR(255))
BEGIN

DECLARE id INT DEFAULT 0;
DECLARE value TEXT;
DECLARE occurance INT DEFAULT 0;
DECLARE i INT DEFAULT 0;
DECLARE COUNT INT;
DECLARE splitted_value VARCHAR(255);
DECLARE done INT DEFAULT 0;
DECLARE cur1 CURSOR FOR SELECT distinct
    SUBSTR(genres,INSTR(genres,'[')+1,INSTR(genres,']') -(1+INSTR(genres,'[')))
                                    FROM spotify_staging.data_w_genres
                                    WHERE genres != '[]';
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN cur1;
  read_loop: LOOP
    FETCH cur1 INTO value;
```

```
    IF done THEN
      LEAVE read_loop;
    END IF;

    SET occurance = (SELECT LENGTH(value) - LENGTH(REPLACE(value, bound, '')) + 1);
    SET i=1;
    WHILE i <= occurance DO
      SET splitted_value = (SELECT LTRIM(REPLACE(SUBSTRING(SUBSTRING_INDEX(value, bound, i),
            LENGTH(SUBSTRING_INDEX(value, bound, i - 1)) + 1), ',', '')));
      SET COUNT = (SELECT COUNT(*) FROM dim_genre WHERE name=splitted_value);
      IF COUNT = 0 THEN
          INSERT INTO dim_genre (name) VALUES (splitted_value);
      END IF;
      SET i = i + 1;
    END WHILE;
  END LOOP;
CLOSE cur1;
END;
|
DELIMITER ;
```

Código 3: *Procedure* para separar os valores presentes nas listas de géneros

4.3 Carregamento

Na fase de carregamento o objetivo era, a partir da nossa *staging area*, carregar os dados corretamente para o nosso *data warehouse*. Para isso seguimos a seguinte metodologia (em passos):

1. Carregamento dos dados das dimensões independentes (tais como: *dim_date*, *dim_key*, *dim_duration*, *dim_genre*, *dim_liveness*, entre outras);
2. Carregamento das dimensões que dependem de outras dimensões (*dim_track_detail* e *dim_artist_detail*);
3. Carregamento das tabelas de facto (*fact_track* e *fact_artist*);
4. Carregamento das tabelas que relacionam músicas com artistas e artistas com múscias (*track_artist* e *artist_genre*).

O trecho de código 4 representa a nossa '*pipeline*' principal de carregamento de dados para o **Data Warehouse**.

5 Sistema de *Business Intelligence*

Como já foi referido, o sistema de *Business Intelligence* foi desenvolvido na plataforma *Tableau Desktop* [4].

A perspetiva de produtora musical tem como objetivo recolher as diferentes métricas do *dataset* e conseguir descobrir relações entre a popularidade de cada música, dos artistas na plataforma face a várias caracteristicas, tal como: qualidade de som, valence (indica a emoção de uma dada música), energia, etc, criando indicadores relevantes e com informação útil.

Com estas comparações e com a devida interpretação dos dados uma produtora conseguia compreender quais as características de uma dada música para que esta seja um sucesso.. Compreender as *trends* que ocorrem ao longo dos anos ajudam a entender a evolução do estilo musical.

Vários tipos de gráficos foram usados para a criação dos indicadores de forma a mostrar relações entre as diferentes métricas. Entre estes podemos encontrar gráficos de barras visto que são de fácil interpretação e gráficos de setores também pela familiaridade que existe com estes.

Desta análise resultou:

- Artistas com maior número de músicas;
- Anos mais populares;
- Artistas mais populares;
- Número de artistas populares por década;
- Emoções apresentadas nas músicas de 1929 vs 2020;
- Artistas pop populares por duração;
- Géneros de músicas mais populares;
- Rácio entre qualidade de som e popularidade;
- Rácio entre energia e nível acústico;

- Relação entre a emoção e a sua capacidade de dançar;
- Músicas rock populares e a sua energia;

Ao analisar as perguntas propostas pretendemos descobrir relações entre os valores utilizados e concluir se há alguma relação entre estes e o sucesso de uma música/artista.

5.1 Artistas com maior número de músicas

Entender quais os artistas que tinham composto o maior número de músicas ao longo da sua carreira foi um indicador criado.

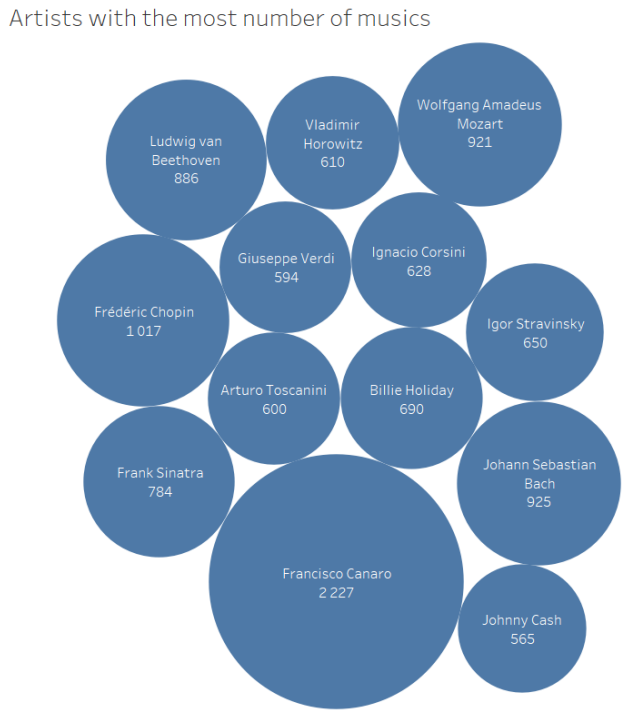

Figura 5: Número de músicas por artista

Para se conseguir mostrar estes artistas foi filtrado um número de músicas por artistas (estes terão de ter 560 músicas mínimo). Esta distribuição ajuda a

compreender que músicos podem ter causado mais impacto na sociedade ao longo dos anos. Denota-se que Francisco Canaro é o artista com mais músicas no *Spotify*.

5.2 Anos mais Populares

Foi investigado que anos detêm mais músicas de sucesso, pelo que foi criado um top 20 com a média da popularidade das músicas de cada ano, sendo que o nível de popularidade vai de 0 a 5.

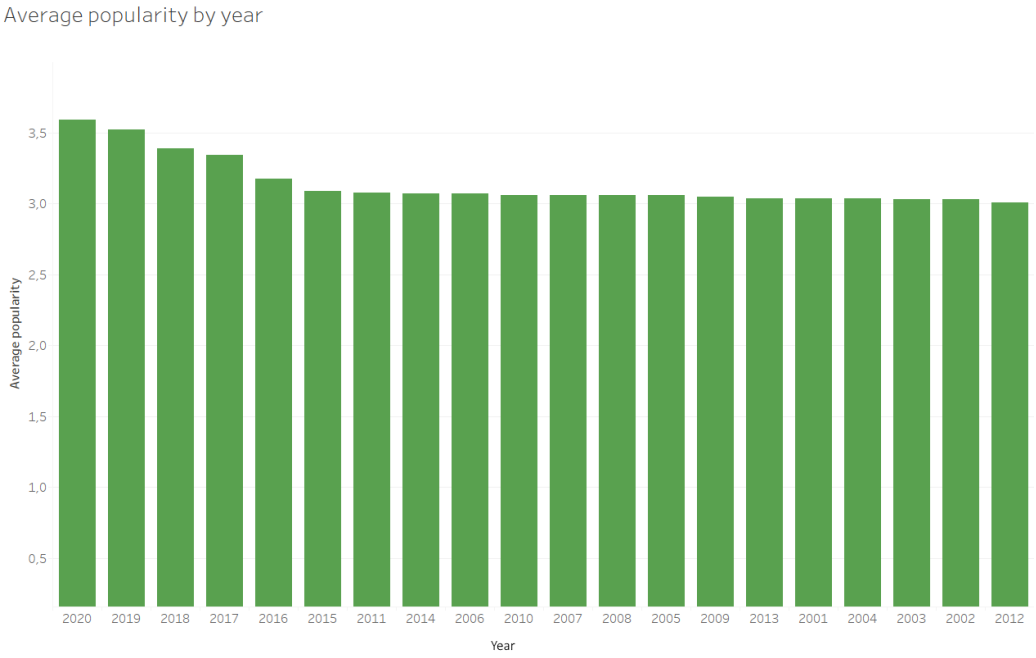

Figura 6: Anos Mais Populares

A figura 6 ajuda a entender qual a prioridade a dar aos dados recebidos de um certo ano. Quanto mais popular o ano, mais facilmente se consegue reconhecer as *trends* ligadas a ele. Isto deve-se ao facto de haver picos de popularidade em músicas é com o surgimento de um novo movimento. Consoante o impacto desse movimento, maior a probabilidade de ele estar numa posição mais elevada na figura 6.

O ano 2020 é o ano com maior popularidade musical. Compreende-se que

com o passar dos anos a produção musical evoluiu e tornou-se mais eficaz a criar músicas com grande popularidade.

5.3 Artistas mais Populares

Os artistas mais populares nesta plataforma de streaming indica quais têm mais impacto atualmente. A média da popularidade de cada artista foi obtida e escolheu-se aqueles cuja média fosse 5 (mais popular). Com esta filtragem obtivemos o top 9 dos artistas.

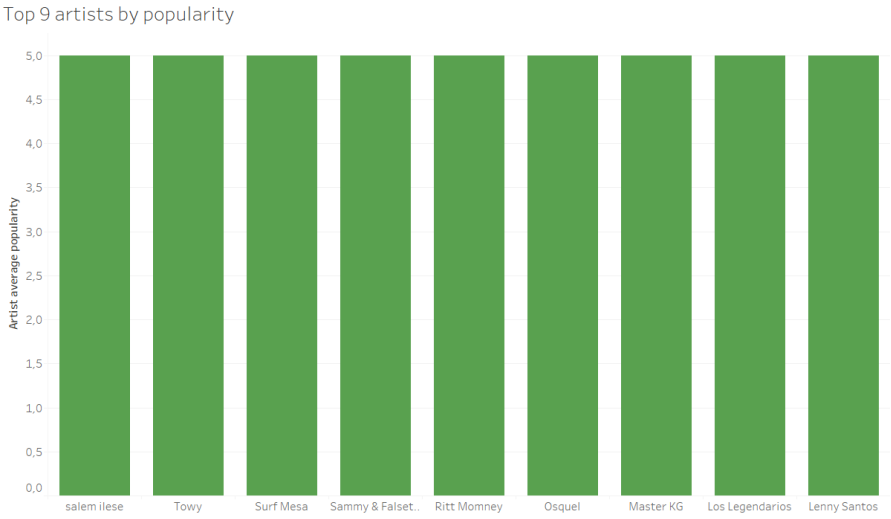

Figura 7: Artistas mais populares

Na figura 7 consegue-se ver que *Salem Ilese, Towy, Surf Mesa, Sammy & Falsetto, Ritt Momney, Osqel, Master KG, Los Legendarios* e *Lenny Santos* foram os 9 artistas mais populares com mais *streams* no *Spotify*.

5.4 Número de Artistas Populares por Década

Como já foi referido anteriormente, encontrar *trends* também é importante. Desta análise resultou uma métrica que nos indica, por década (visto que cada

década também pode ser identificada por *trends* que a marcam), a quantidade de artistas que ficaram populares.

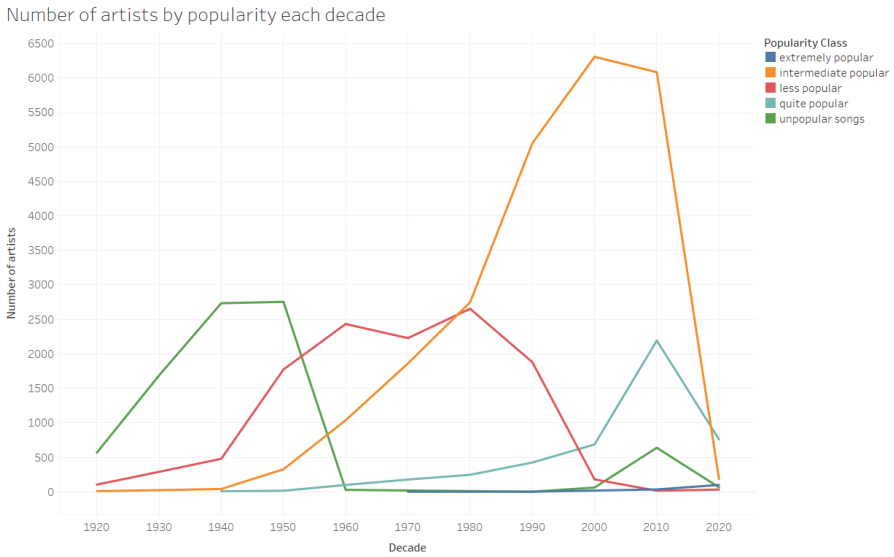

Figura 8: Número Artistas Populares por Década

A Figura 8 ajuda a compreender quais as décadas que marcaram mais a sociedade. Tirando partido desta informação, pode-se focar mais em certas décadas para encontrar não só elementos presentes nas músicas mas também encontrar uma possível relação com a sua popularidade para poder tentar replicar um efeito semelhante com a música a ser criada. Como se pode ver, os anos 2000 foram dos anos com maior popularidade, dado que tem um grande número de músicas com popularidade intermédia, e um reduzido número de músicas com popularidade baixa.

5.5 1929 vs 2020

Para se criar um termo de comparação dentro do estado de espirito de uma música foi-se procurar dois anos marcados por algo que mudou por completo a

perspetiva da humanidade. Essa perspetiva mudou na música? Quais as diferenças?

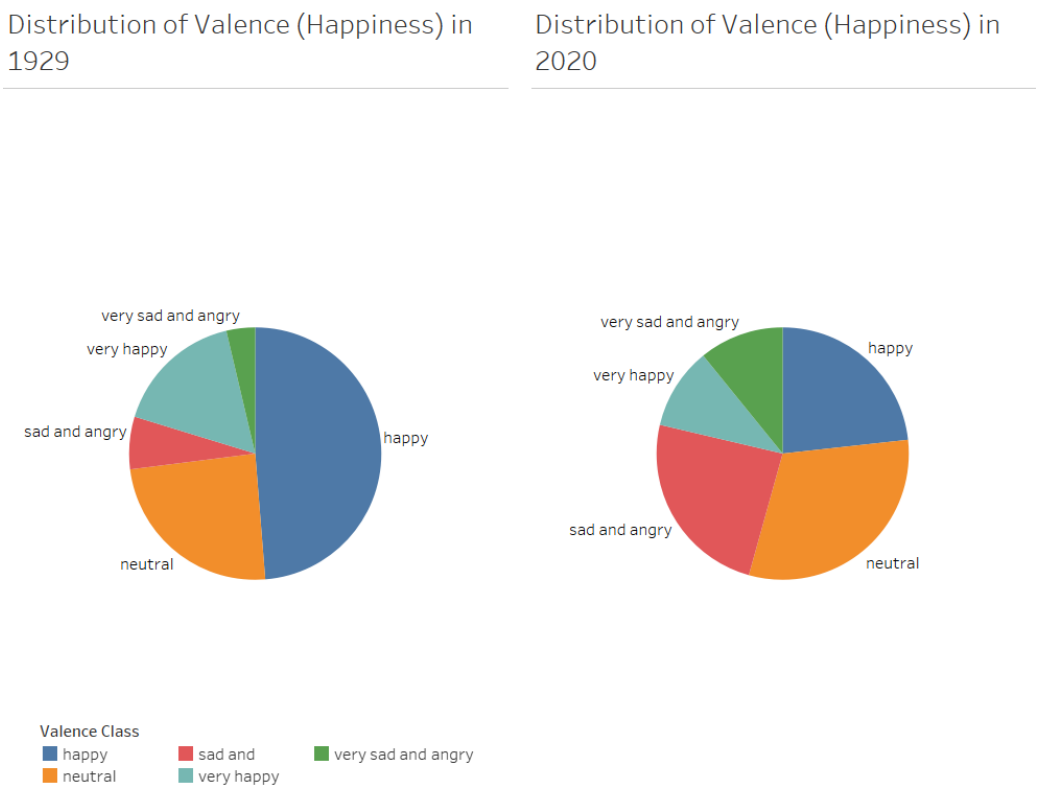

Figura 9: Valence na queda da bolsa (1929) e valence em 2020 (pandemia)

A figura 9 apresenta os anos 1929 e 2020. 1929 foi marcado pela queda da bolsa de *Wall Street* e, mais recentemente, o ano 2020, marcado pelo surgimento da pandemia relacionada ao COVID-19. Procurou-se encontrar uma relação entre o negativismo e as situações infelizes de cada ano. Conclui-se que, no geral, o ano 2020 apresenta uma distribuição diferente comparativamente ao ano 1929. Foram produzidas mais músicas *sad and angry* e *very sad and angry* e menos músicas *happy* e *very happy*. Seria esperado que ambas apresentassem baixos níveis de felicidade e uma fatia maior de músicas com tom mais triste. Pelo contrário, o ano 1929 foi marcado por músicas com uma entoação mais positiva. O ano 2020 foi

aquele que foi de encontro às expectativas criadas pelo grupo, talvez pelo facto de ter sido uma situação que o grupo presenciou.

5.6 Artistas Pop Populares por Duração

Cada artista é diferente, mas quando lançam uma música com o mesmo género tem de haver algo que os distinga. Com esta métrica pretende-se procurar uma relação entre a popularidade dos artistas pop com a duração da sua música para saber o melhor intervalo de tempo a utilizar numa música.

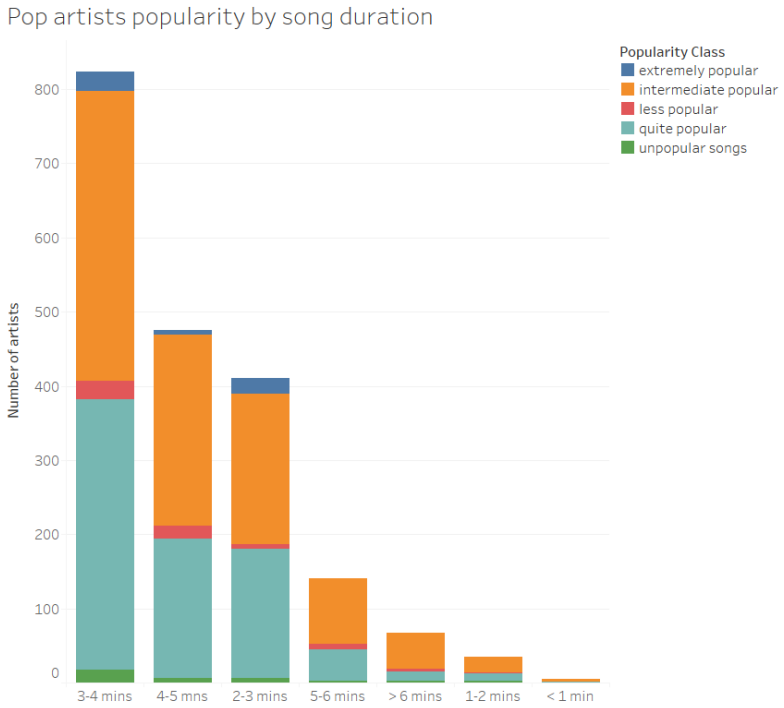

Figura 10: Artistas Pop populares por duração

Podemos observar na figura 10 que as músicas extremamente populares estão entre os 2-3 minutos e os 3-4 minutos, em comparação o número de músicas menos populares têm uma grande predisposição além da anterior incluindo os 4-5 minutos, assim, pode-se concluir que para haver uma maior chance de sucesso, deve-se escolher uma música entre os 2-4 minutos, visto que é nesta gama que, de um

modo geral, se situam as músicas mais populares, salvo algumas exceções como se pode ver no indicador.

5.7 Géneros de Música Mais Populares

Tentou-se perceber quais os géneros musicais mais ouvidos pelos utilizadores de *Spotify*. Para isso foi usada a média da popularidade de cada género, com o objetivo de se perceber realmente os géneros mais influentes.

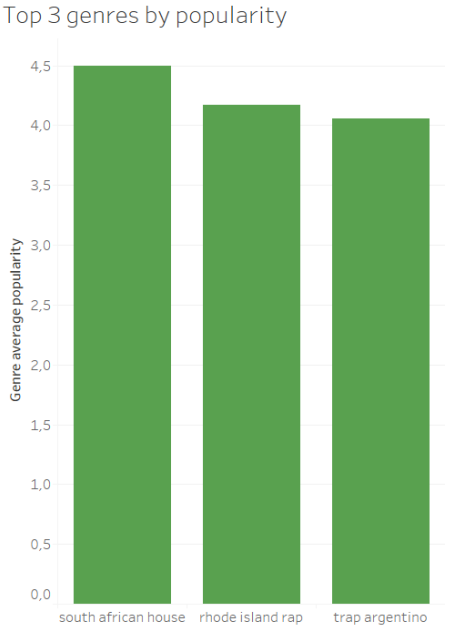

Figura 11: Top 3 Géneros mais Populares

Com esta filtragem chegou-se então á conclusão que os géneros com uma maior média de popularidade presentes, no *Spotify*, são *south african house*, *rhode island rap* e *trap argentino*. Este indicador não vai de encontro a possíveis expetativas iniciais, possivelmente pelo facto de existir uma grande variabilidade dos géneros (múltiplos géneros de *pop* diferentes, o mesmo no rock, *country*, etc), contudo, permite perceber que dentro dos géneros mais populares, na atualidade, estão: *house* e *rap*

5.8 Rácio entre Qualidade de Som e Popularidade

A qualidade de som afeta a popularidade? Esta questão foi colocada com o objetivo de entender se realmente existe um impacto na popularidade quando a qualidade de som não é a melhor. A razão pela qual este indicador foi criado deve-se ao facto de muitas músicas ocorrerem em períodos onde o nível de produção não é tão elevado quanto os anos mais recentes.

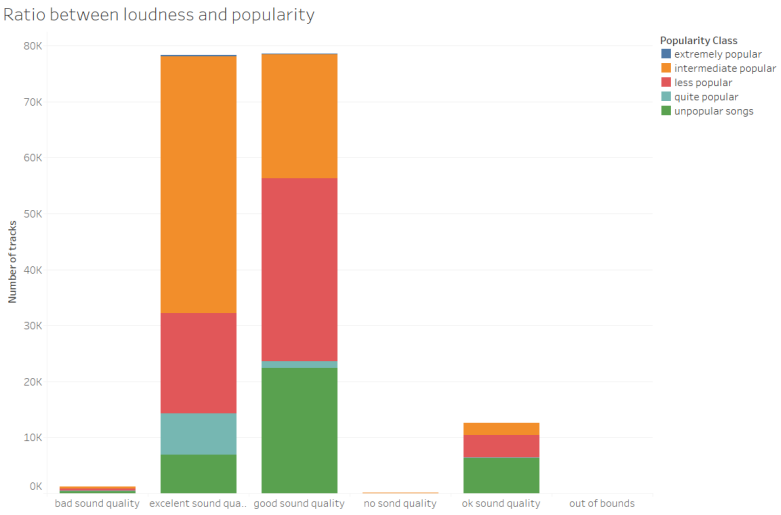

Figura 12: Rácio entre Qualidade de Som e a Popularidade

Como a figura 12 indica, quanto melhor qualidade de som, maior a probalilidade de esta estar compreendida entre um nível 3 a 5 de popularidade (popularidade intermédia a extremamente popular). Quanto pior a qualidade de som menos probabilidade tem de se tornar popular.

5.9 Rácio entre energia e Nível Acústico

Para este indicador a questão 'Quanto mais energética a música menos acoustica é?' foi colocada.

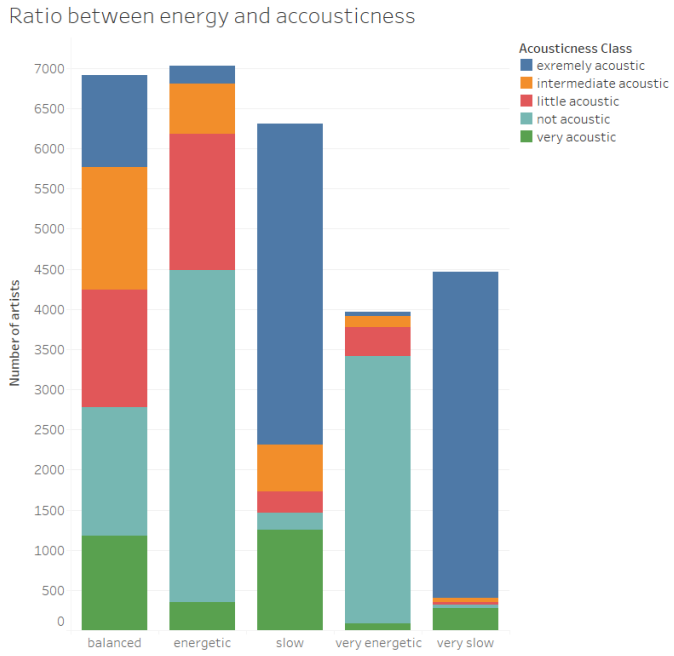

Figura 13: Rácio entre energia e accousticness

A figura 13 mostra que músicas mais lentas têm tendência a ter mais elementos acústicos do que quando uma música é energética. Isto é, a tendência mostra que subindo na escala de energia cada vez menos acústica a música se torna. Ajuda a compreender se realmente a energia pode e deve ser afetada pelo nivel de acústico e o quão normal e proporcional esta relação é.

5.10 Relação entre a Emoção e a Capacidade de Dançar

Procurou-se encontrar uma relação entre a parte sentimental de uma música e a sua capacidade de fazer com que as pessoas dançem ao som da mesma.

Poder dançar ao som de uma música é algo importante em certas ocasiões. Por exemplo, no caso de um concerto, que músicas são indicadas a colocar quando objetivo é involver a multidão?

Para isso usou-se dois *pie charts* para poder haver um termo comparativo

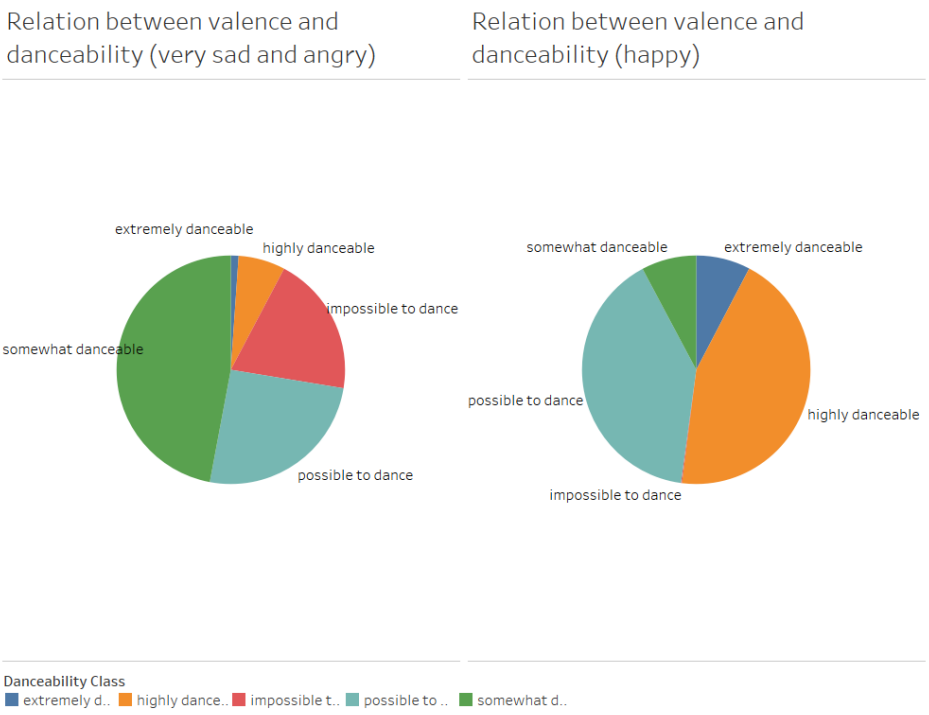

Figura 14: Relação entre Emoção e a sua Capacidade de Dançar

entre dois sentimentos completamente distintos: *very sad and angry* e *happy*. Conclui-se que uma música com emoções positivas têm uma taxa bastante maior de capacidade de dançar do que as músicas mais péssimistas.

5.11 Músicas Rock Populares e a Sua Energia

Alguns tipos de música estão associados a certos tipos de energia. Normalmente quando pensamos numa música eletrónica estamos à espera de algo com altos níveis de energia, enquanto que ao ouvir músicas mais calmas a energia desta seja, de forma proporcional, mais baixa. Tomando este princípio, decidiu-se investigar o tipo de energia que é esperado estar presente numa música de Rock.

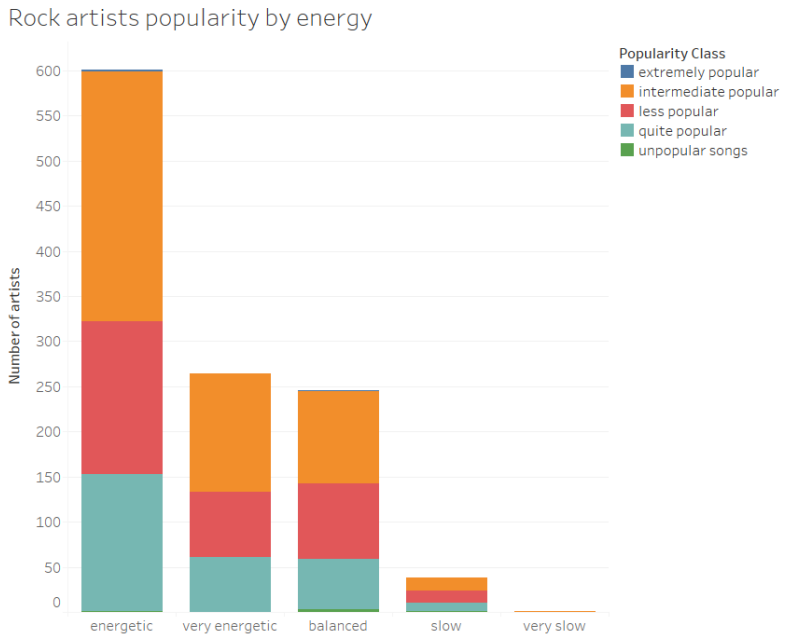

Figura 15: Relação entre a popularidade de uma música Rock e a sua energia

Observa-se que a grande parte das músicas de rock são energéticas, enquanto que há poucas que são mais paradas, quando juntamos este aspeto à popularidade das mesmas podemos observar que a melhor opção é realmente compor uma música Rock energética. Em termos de rácio, a popularidade da música tem a tendência a diminuir com níveis de energias mais baixos. Apesar de ser o resultado esperado, houve a curiosidade de saber se, com menos energia, seria possível uma música Rock se tornar popular. Algo a denotar foi que mesmo com o devido equilibrio em termos energéticos esta pode se tornar quase tão popular como uma música muito energética. Aliás, existem quase tantas músicas extremamente populares quando a energia é equilibrada e quando é energética. Indica assim que ao compor músicas rock, nem sempre uma música mais energética significa que será mais popular.

6 Conclusão

Para o processo de modelação foi adotada a perspetiva do utilizador final (isto é, uma produtora músical ou uma empresa com forte presença na indústria da música) e foram escolhidas as dimensões e os factos a incluir no *Data Warehouse*. Este processo foi feito de forma a responder às seguintes perguntas:

- Onde e como é que são recolhidos os dados?
- Qual deve ser a granularidade do *Data Warehouse*?
- Que dimensões devem existir?
- Que medidas devem ser incluidas nos factos?

Após termos definido isto, passou-se para o processo de implementação do *Data Warehouse*, seguindo a metodologia definida por *Ralph Kimball* em [6], através da implementação do schema oriundo do processo de modelação e da implementação dos processos de ETL (Extração, Transformação e Carregamento).

A fase final do trabalho foi, através do *Tableau*, criar indicadores que permitissem tirar conclusões sobre os dados e inferir regras e padrões que sejam úteis ao processo de decisão.

Deste modo, foi possível compreender melhor a arquitetura de um *Data Warehouse*, conseguir identificar um *Dataset* adequado e a partir do mesmo fazer uma modelação dimensional, importar os respetivos dados e popular todas as tabelas criadas com a modelação através de comandos *SQL*, e, por fim, fazer a devida exploração e conseguir obter uma boa análise de dados com o manuseamento da ferramenta *Tableau*.

O grupo considera ter cumprido os requisitos do projeto com sucesso e que o trabalho desenvolvido reflete não só o trabalho desenvolvido no decorrer da disciplina, mas também os objetivos delineados para o mesmo de uma forma fidedigna.

A Carregamento dos dados no *Data Warehouse*

```
CALL populate_dim_date('1921-01-01','2020-12-31');
CALL populate_dim_genre(',');

-- dim_key
insert into dim_key (id_key, key_class)
select key_id, key_value
from spotify_staging.key_values;

-- dim_duration
insert into dim_duration (id_duration, duration_class) values
    (1, '< 1 min'),
    (2, '1-2 mins'),
    (3, '2-3 mins'),
    (4, '3-4 mins'),
    (5, '4-5 mns'),
    (6, '5-6 mins'),
    (7, '> 6 mins');

-- dim_valence
insert into dim_valence (id_valence, valence_class) values
    (1, 'very sad and angry'),
    (2, 'sad and angry'),
    (3, 'neutral'),
    (4, 'happy'),
    (5, 'very happy');

-- dim_liveness
insert into dim_liveness (id_liveness, liveness_class) values
    (1, 'not live concert'),
    (2, 'maybe live concert'),
    (3, 'probably live concert'),
    (4, 'live concert');

-- dim_loudness
insert into dim_loudness (id_loudness, loudness_class) values
    (1, 'no sond quality'),
    (2, 'bad sound quality'),
    (3, 'ok sound quality'),
    (4, 'good sound quality'),
    (5, 'excelent sound quality'),
    (6, 'out of bounds');

-- dim_acousticness
insert into dim_acousticness (id_acousticness, acousticness_class) values
    (1, 'not acoustic'),
    (2, 'little acoustic'),
    (3, 'intermediate acoustic'),
    (4, 'very acoustic'),
```

```
    (5, 'exremely acoustic');

-- dim_popularity
insert into dim_popularity (id_popularity, popularity_class) values
    (1, 'unpopular songs'),
    (2, 'less popular'),
    (3, 'intermediate popular'),
    (4, 'quite popular'),
    (5, 'extremely popular');

-- dim_speechiness
insert into dim_speechiness (id_speechiness, speechiness_class) values
    (1, 'instrumental "only"'),
    (2, 'blend of music and vocals/words'),
    (3, 'vocals/words "only"');

-- dim_instrumentalness
insert into dim_instrumentalness (id_instrumentalness, instrumentalness_class) values
    (1, 'very little instrumental'),
    (2, 'some instrumental'),
    (3, 'perfect blend'),
    (4, 'very instrumental'),
    (5, 'extremely instrumental');

-- dim_energy
insert into dim_energy (id_energy, energy_class) values
    (1, 'very slow'),
    (2, 'slow'),
    (3, 'balanced'),
    (4, 'energetic '),
    (5, 'very energetic');

-- dim_danceability
insert into dim_danceability (id_danceability, danceability_class) values
    (1, 'impossible to dance'),
    (2, 'somewhat danceable'),
    (3, 'possible to dance'),
    (4, 'highly danceable'),
    (5, 'extremely danceable');

-- dim_track_detail
insert into dim_track_detail (name, tempo, id_duration, id_key, id_spotify_track)
select name, CAST(tempo as float) as tempo,
    duration_classification(duration_ms), `key`, id
from spotify_staging.data;

-- dim_artist_detail
insert into dim_artist_detail (name, tempo, id_avg_duration, id_avg_key)
select artists, tempo, duration_classification(duration_ms), `key`
```

```
from spotify_staging.data_w_genres;

-- fact_track
insert into spotify.fact_track (`mode`, explicit, id_release_date, id_valence,
    id_details, id_acousticness, id_danceability, id_energy, id_instrumentalness,
    id_liveness, id_loudness, id_popularity, id_speechiness)
select `mode`, explicit, t1.id_date, valence_classification(valence), t2.id_details,
    acousticness_classification(acousticness), danceability_classification(danceability),
    energy_classification(energy), instrumentalness_classification(instrumentalness),
    liveness_classification(liveness), loudness_classification(loudness),
    popularity_classification(popularity), speechiness_classification(speechiness)
from spotify_staging.data
    inner join dim_date t1 on t1.date = STR_TO_DATE(data.release_date, '%Y-%m-%d')
    inner join dim_track_detail t2 on t2.id_spotify_track = data.id;

-- fact_artist
insert into spotify.fact_artist (`mode`, id_valence, id_details, id_acousticness,
    id_danceability, id_energy, id_instrumentalness, id_liveness, id_loudness,
    id_popularity, id_speechiness)
select `mode`, valence_classification(valence), t1.id_details,
    acousticness_classification(acousticness), danceability_classification(danceability),
    energy_classification(energy), instrumentalness_classification(instrumentalness),
    liveness_classification(liveness), loudness_classification(loudness),
    popularity_classification(popularity), speechiness_classification(speechiness)
from spotify_staging.data_w_genres
    inner join dim_artist_detail t1 on t1.name = data_w_genres.artists;

CALL populate_track_artist(',');
CALL populate_artist_genre(',');
```

Código 4: Carregamento dos dados no *Data Warehouse*

Referências

[1] *Magic Quadrant for Analytics and Business Intelligence Platforms*, "https://www.gartner.com/doc/reprints?id=1-3TXXSLV&ct=170221&st=sb&ocid=mkto_eml_EM597235A1LA1", Acedido: 01-01-2021.

[2] *Spotify Dataset 1921-2020, 160k+ Tracks*, "https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks", Acedido: 02-01-2021.

[3] *PowerBI*, "https://powerbi.microsoft.com/pt-pt/", Acedido: 16-01-2021.

[4] *Tableau Desktop*, "https://www.tableau.com/products/desktop", Acedido: 09-01-2021.

[5] *ETL Tool Architecture - Data Warehouse ETL Toolkit*, "https://www.wisdomjobs.com/e-university/data-warehouse-etl-toolkit-tutorial-201/etl-tool-architecture-8029.html", Acedido: 16-01-2021.

[6] R. Kimball e M. Ross, *The Data Warehouse Toolkit*, 3ª ed., Acedido: 14-01-2021.