



**Bases de Dados (LEI)**  
**2024/2025 – 2º Semestre**

**Relatório Trabalho Prático**  
**Aplicação Gestora de Músicas**  
**2ª Fase**  
**Servidor local**

**Realizado pelo Grupo 51 da Prática 5:**  
**André Marques Tavares 67161**  
**Guilherme Miguel Henriques Martins Ferreira Rocha 69112**  
**Miguel Castanho Barbosa 68206**

# Índice

1. Descrição da aplicação gestora de músicas.....	3
2. Diagrama .....	4
2.1. Modelo ER.....	4
2.1.1. Alterações relativamente à 1ª Fase .....	4
2.2. Modelo Relacional.....	5
3. Código SQL .....	7
3.1. Raciocínio para a criação dos Triggers .....	15
4. Limitações da Base de Dados.....	17
5. Consultas interessantes .....	17
6. Funcionalidades .....	19
6.1. Páginas de entrada .....	19
6.2. Atributos de fácil compreensão.....	19
6.3. Valores derivados.....	20
6.4. Atualização da base de dados .....	20
6.5. Lista de valores (LoV) .....	20
6.6. Breadcrumbs .....	20
6.7. Relatórios interligados.....	20
6.8. Detalhe condicional .....	21
6.9. Master-detail.....	21
6.10. Consultas interessantes.....	21
6.11. Início da subscrição paga.....	21
7. Guia da Aplicação .....	22
7.1. Página inicial .....	22
7.2. Atributos de fácil compreensão.....	22
7.3. Valores derivados.....	22
7.4. Atualização da base de dados .....	22
7.5. Lista de valores (LoV) .....	23
7.6. Breadcrumbs .....	23
7.7. Relatórios interligados.....	23
7.8. Detalhe condicional .....	23
7.9. Master-detail.....	23
7.10. Consultas interessantes.....	23
7.11. Início da subscrição paga.....	24

# 1. Descrição da aplicação gestora de músicas

A base de dados foi desenvolvida para suportar uma aplicação de música digital semelhante ao Spotify, gerindo contas de utilizadores, subscrições, artistas, músicas, álbuns, playlists e pagamentos. A criação de uma subscrição ou de um perfil de artista exige obrigatoriamente a existência de uma conta. Os dados associados a essa conta permanecem ativos nas entidades relacionadas (como subscrições e artistas) enquanto a conta existir; caso a conta seja eliminada, todas as suas ligações e dados associados são igualmente removidos.

As subscrições podem ser gratuitas ou pagas. Apenas utilizadores com uma conta podem criar uma subscrição paga. Estas subscrições podem ser inicialmente criadas sem a realização imediata de um pagamento, mas só serão ativadas quando um pagamento válido for efetuado. Uma subscrição paga pode ser estendida a qualquer momento mediante um novo pagamento, e o seu tipo (Premium ou Family) pode ser alterado enquanto a subscrição atual estiver em vigor.

As subscrições pagas são obrigatoriamente do tipo Premium ou Family. A subscrição Premium é individual e associada a uma única conta, enquanto a subscrição Family permite agregar até cinco contas diferentes. No entanto, estas contas associadas não podem estar previamente ligadas a qualquer outra subscrição paga. Cada subscrição paga está vinculada a um dos pacotes disponíveis, com durações de 1, 3, 6 ou 12 meses. Os preços dos pacotes Premium são: 5,99 € (1 mês), 17,07 € (3 meses), 31,63 € (6 meses) e 57,50 € (12 meses). Já os pacotes Family custam 14,99 € (1 mês), 42,72 € (3 meses), 79,15 € (6 meses) e 143,90 € (12 meses). Se o valor do pagamento efetuado não corresponder exatamente ao valor do pacote selecionado, o pagamento é recusado e a subscrição não é ativada nem prorrogada. Uma subscrição paga só pode ser removida da base de dados quando estiver expirada, ou seja, quando a soma da data de início com a duração do pacote ultrapassar a data atual.

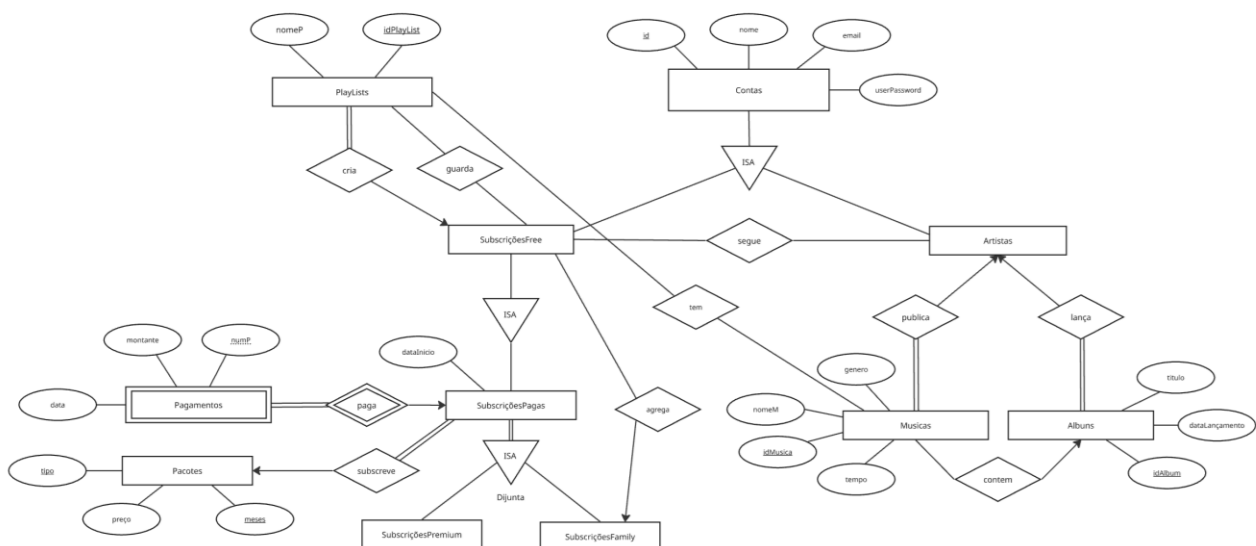
Em relação ao conteúdo musical, apenas artistas podem lançar músicas. Cada música possui um género, um identificador único atribuído automaticamente (ID da música), um nome e a duração (tempo). As músicas podem fazer parte de um álbum ou ser lançadas individualmente, mas se fizerem parte de um álbum, todas devem ser da autoria do mesmo artista, não sendo permitido misturar músicas de diferentes artistas no mesmo álbum.

Cada álbum representa um conjunto de músicas agrupadas num lançamento feito por um ou mais artistas. Um álbum inclui um título, a data de lançamento e um ID único atribuído automaticamente e aleatoriamente.

Os utilizadores da aplicação podem criar playlists personalizadas, seleccionando livremente músicas de diferentes artistas. Cada playlist tem um nome e um identificador único atribuído automaticamente e aleatoriamente. Além de criar playlists, os utilizadores também podem pesquisar playlists criadas por outros utilizadores e adicioná-las à sua própria lista de playlists, permitindo uma experiência mais social e colaborativa de descoberta musical.

## 2. Diagrama

### 2.1. Modelo ER



#### 2.1.1. Alterações relativamente à 1ª Fase

Tirámos atributos redundantes da Playlists e de Aluns. Mudámos as relações publica e lança para muitos para um pois achamos que fazia mais sentido no contexto do nosso projeto considerar que uma Música deve ter um só Artista e o mesmo para os Aluns. Adicionamos a entidade Pacotes para facilitar a identificação da Subscrição. Alteramos o conceito das Subscrições, passando agora a ter uma entidade geral para

SubscriçõesPagas, que tem um ISA Disjunto para SubscriçõesPremium e SubscriçõesFamily. Tivemos a iniciativa de alterar o conceito das Subscrições para permitir que existam SubscriçõesFamily sem que tenham que obrigatoriamente tar presentes nas SubscriçõesPremium primeiro.

## 2.2. Modelo Relacional

Aqui encontram-se todas as tabelas com as suas chaves primarias e estrangeiras.

### Tabelas Principais

Contas (id, nome, email, userPassword)

SubscriçõesFree(id)

- id e chave estrangeira de Contas

Artistas (id)

- id e chave estrangeira de Contas

SubscriçõesPagas (id, meses, tipo, dataInicio)

- id e chave estrangeira de SubscriçõesFree
- meses e chave estrangeira de Pacotes
- tipo e chave estrangeira de Pacotes

SubscriçõesFamily(id)

- id e chave estrangeira de SubscriçõesPagas

SubscriçõesPremium(id)

- id e chave estrangeira de SubscriçõesPagas

Pagamentos (id, numP, montante, data)

- id e chave estrangeira de SubscriçõesPagas

PlayLists (idPlayList, id, nomeP)

- id e chave estrangeira de SubscriçõesFree

Musicas (idMusica, id, nomeM, tempo, genero)

- id e chave estrangeira de Artistas

Albums (idAlbum, id, titulo, dataLançamento)

- id e chave estrangeira de Artistas

Pacotes (meses, tipo, preço)

segue (id.SubscriçõesFree, id.Artistas)

- id.SubscriçõesFree é chave estrangeira de SubscriçõesFree
- id.Artistas é chave estrangeira de Artistas

guarda (id, idPlayList)

- id é chave estrangeira de SubscriçõesFree
- idPlayList é chave estrangeira de PlayLists

tem (idPlayList, idMusica)

- idPlayList é chave estrangeira de PlayLists
- idMusica é chave estrangeira de Musica

agrega (id.SubscriçõesFree, id.SubscriçõesFamily)

- id.SubscriçõesFree é chave estrangeira de SubscriçõesFree

- id.Family é chave estrangeira de SubscriçõesFamily

contém(idMusica, idAlbum)

- idAlbum é chave estrangeira de Albuns
- idMusica é chave estrangeira de Musicas

### 3. Código SQL

```
CREATE OR REPLACE FUNCTION account_num_pag (p_id IN NUMBER)
RETURN INTEGER IS pag_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO pag_count
    FROM Pagamentos
    WHERE Pagamentos.id = p_id;
    RETURN pag_count + 1;
END account_num_pag;
/
```

```
CREATE OR REPLACE TRIGGER trg_insert_subscription_type_after
AFTER INSERT ON SubscicoesPagas
FOR EACH ROW
BEGIN
    IF :NEW.tipo = 'Premium' THEN
        INSERT INTO SubscicoesPremium
```

```

VALUES (:NEW.id);
ELSE
    INSERT INTO SubscricoesFamily
    VALUES (:NEW.id);
END IF;
END;
/

```

```

CREATE OR REPLACE TRIGGER trg_validate_payment_amount
BEFORE INSERT ON Pagamentos
FOR EACH ROW
DECLARE
    v_expected_amount DECIMAL(5,2);
    num_pag number;
BEGIN
    SELECT preco INTO v_expected_amount
    FROM SubscricoesPagas NATURAL JOIN Pacotes
    WHERE SubscricoesPagas.id = :NEW.id;

    IF :NEW.montante != v_expected_amount THEN
        RAISE_APPLICATION_ERROR(-20100, 'Montante inválido. Esperado: ' ||
v_expected_amount);
    END IF;

    :new.numP := account_num_pag(:NEW.id);
    :new.data := SYSDATE;

```



```

UPDATE SubscicoesPagas
SET dataInicio = SYSDATE
WHERE id = :NEW.id;
END;
/

```

```

CREATE OR REPLACE TRIGGER trg_insert_subscription_type_before
BEFORE INSERT ON SubscicoesPagas
FOR EACH ROW
DECLARE
    v_count1 NUMBER;
    v_count2 NUMBER;
    v_count3 NUMBER;
    v_amount_needed DECIMAL(5,2);
BEGIN
    SELECT COUNT(*) INTO v_count1 FROM agrega WHERE id_SubscicoesFree =
:NEW.id;

    IF v_count1 > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'Account already in family');
    END IF;

    SELECT COUNT(*) INTO v_count2 FROM SubscicoesFree WHERE id = :NEW.id;

    IF v_count2 = 0 THEN
        INSERT INTO SubscicoesFree VALUES (:NEW.id);
    END IF;
END;

```

```

END IF;

SELECT preco INTO v_amount_needed
FROM Pacotes
WHERE meses = :NEW.meses AND tipo = :NEW.tipo;

:NEW.dataInicio := NULL;
END;
/

CREATE OR REPLACE TRIGGER trg_insert_subscription_type_after
AFTER INSERT ON SubscricoesPagas
FOR EACH ROW
BEGIN
    IF :NEW.tipo = 'Premium' THEN
        INSERT INTO SubscricoesPremium
        VALUES (:NEW.id);
    ELSE
        INSERT INTO SubscricoesFamily
        VALUES (:NEW.id);
    END IF;
END;
/

CREATE OR REPLACE TRIGGER trg_remove_subscription
BEFORE DELETE ON SubscricoesPagas
FOR EACH ROW
DECLARE

```

```

v_end_date DATE;

BEGIN

    v_end_date := ADD_MONTHS(:OLD.dataInicio, :OLD.meses);

    IF SYSDATE < v_end_date THEN

        RAISE_APPLICATION_ERROR(-20100, 'Cannot delete active subscription.
Ends on ' || TO_CHAR(v_end_date, 'DD-MON-YYYY'));

    END IF;

END;

/

```

```

CREATE OR REPLACE TRIGGER trg_disj_Premium
BEFORE INSERT ON SubscricoesPremium
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM SubscricoesFamily
    WHERE id = :NEW.id;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'Account cannot be both Premium and
Family');
    END IF;
END;

/

```

```

CREATE OR REPLACE TRIGGER trg_disj_Family

```

```

BEFORE INSERT ON SubscricoesFamily
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM SubscricoesPremium
    WHERE id = :NEW.id;

    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'Account cannot be both Premium and
Family');
    END IF;
END;
/

```

```

CREATE OR REPLACE TRIGGER trg_update_subscription_type
BEFORE UPDATE ON SubscricoesPagas
FOR EACH ROW
BEGIN
    IF :OLD.id != :NEW.id THEN
        RAISE_APPLICATION_ERROR(-20100, 'Id cannot be updated.');
```

```

    END IF;

    IF :OLD.tipo != :NEW.tipo THEN
        IF :OLD.tipo = 'Premium' THEN
            DELETE FROM SubscricoesPremium WHERE id = :OLD.id;
        ELSIF :OLD.tipo = 'Family' THEN
            DELETE FROM SubscricoesFamily WHERE id = :OLD.id;
        END IF;
    
```

```

IF :NEW.tipo = 'Premium' THEN
    INSERT INTO SubscricoesPremium
    VALUES (:NEW.id);
ELSIF :NEW.tipo = 'Family' THEN
    INSERT INTO SubscricoesFamily
    VALUES (:NEW.id);
END IF;
:NEW.dataInicio := NULL;
END IF;
END;
/

```

```

CREATE OR REPLACE TRIGGER trg_album_artist_consistency
BEFORE INSERT OR UPDATE ON contem
FOR EACH ROW
DECLARE
    v_song_artist NUMBER;
    v_album_artist NUMBER;
BEGIN
    SELECT id INTO v_song_artist
    FROM Musicas
    WHERE idMusica = :NEW.idMusica;

    SELECT id INTO v_album_artist
    FROM Albuns
    WHERE idAlbum = :NEW.idAlbum;

```

```

    IF v_song_artist != v_album_artist THEN
        RAISE_APPLICATION_ERROR(-20100, 'Song and album must be by the same
artist');
    END IF;
END;
/

```

```

CREATE OR REPLACE TRIGGER trg_validate_family_account
BEFORE INSERT OR UPDATE ON agrega
FOR EACH ROW
DECLARE
    v_count1 NUMBER;
    v_count2 NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count1
    FROM agrega
    WHERE id_SubscricoesFamily = :NEW.id_SubscricoesFamily;

    SELECT COUNT(*) INTO v_count2
    FROM SubscricoesPagas
    WHERE id = :NEW.id_SubscricoesFree;

```

```

IF v_count1 >= 5 THEN

    RAISE_APPLICATION_ERROR(-20100, 'Family account cannot have more than 5
associated free accounts');

END IF;

IF v_count2 > 0 THEN

    RAISE_APPLICATION_ERROR(-20100, 'This account is already associated with a
paid subscription');

END IF;

END;

/

```

### 3.1. Raciocínio para a criação dos Triggers

*account\_num\_pag :*

Esta função recebe o Id do utilizador, e calcula qual será o número de pagamento seguinte.

*trg\_validate\_payment\_amount:*

Este trigger serve para validar o montante do pagamento. Caso o montante seja válido, atualiza a data do pagamento para a data do sistema, e a data de início da Subscrição Paga que efetuou o pagamento.

*trg\_insert\_subscription\_type\_before:*

Este trigger serve para evitar que uma Subscrição Free já agregada a uma Subscrição Family, consiga ser adicionada às Subscrições Pagas. Caso a Subscrição

Free não exista (e neste caso nunca poderia estar agregada a uma Subscrição Family) é adicionada às Subscrições Free antes de ser adicionada às Subscrições Pagas. A data de início é definida como NULL enquanto não houver um Pagamento.

*trg\_insert\_subscription\_type\_after:*

Este trigger serve para após a conta ser adicionada às Subscrições Pagas, ser adicionada às Subscrições Premium ou Subscrições Family de acordo com o tipo da Subscrição escolhida.

*trg\_update\_subscription\_type:*

Este trigger serve para não permitir atualizações aos Id's das Contas, e no caso de atualização do tipo de Subscrição, remover da Subscrição antiga e adicionar ao novo tipo de Subscrição, caso os tipos sejam diferentes. Caso o tipo seja alterado, a data de início é atualizada para NULL.

*trg\_remove\_subscription:*

Este trigger serve para não permitir a remoção de Subscrições Pagas que ainda não expiraram (data de início + número de meses do Pacote <= data atual).

*trg\_disj\_Premium:*

Este trigger serve para garantir a disjunção das Subscrições Premium e Subscrições Family.

*trg\_disj\_Family:*

Este trigger serve para garantir a disjunção das Subscrições Family e Subscrições Premium.

*trg\_validate\_family\_account:*

Este trigger serve para ao agregar uma Conta a uma Subscrição Family, garantir que há espaço para a sua adição (número de contas agregadas < 5) e que a Conta a ser agregada não está associada a uma Subscrição Paga.



*trg\_album\_artist\_consistency:*

Este trigger serve para garantir que as Músicas contidas num Album são todas do Artista criador do Album.

## 4. Limitações da Base de Dados

A nossa maior limitação é o não ter como guardar o histórico de Pagamentos de uma conta que seja removida das Subscrições Pagas, pois para que isso fosse possível seria necessário manter a conta nas Subscrições Pagas mesmo que a Subscrição não estivesse em vigor pois os Pagamentos precisam do id da Subscrição.

Outra limitação surgiu após uma alteração necessária nas relações lança entre Artistas e Musicas e publica entre Artistas e Albuns, onde tivemos que limitar a que para cada Album e cada Música só possa estar diretamente associado um Artista, pois caso contrário, dificultava consideravelmente a implementação da Base de Dados.

E por último mas não menos importante, o não ser possível inserir diretamente nas SubscriçõesPremium e SubscriçõesFamily pois é necessário que exista uma SubscriçãoPaga para que as possamos criar.

## 5. Consultas interessantes

```
SELECT c.id, c.nome, c.email  
FROM Contas c  
INNER JOIN SubscicoesFree f ON c.id = f.id  
WHERE f.id NOT IN (SELECT id FROM SubscicoesPagas)  
AND f.id NOT IN (SELECT id_SubscicoesFree FROM agrega);
```

A pesquisa acima separa as SubscriçõesFree das que estão associadas a algum tipo de SubscriçõesPagas.

```

SELECT
    C.nome AS "Artista",
    COUNT(S.id_SubscicoesFree) AS "Número de Seguidores"
FROM segue S
INNER JOIN Artistas A ON S.id_Artistas = A.id
INNER JOIN Contas C ON A.id = C.id
GROUP BY C.nome
ORDER BY COUNT(S.id_SubscicoesFree) DESC;

```

A pesquisa acima devolve o número de pessoas que segue um Artista.

```

SELECT id, nome, email
FROM (
    SELECT cf.id, cf.nome, cf.email
    FROM SubscicoesFamily sf
    INNER JOIN Contas cf ON cf.id = sf.id

    UNION

    SELECT f.id, f.nome, f.email
    FROM SubscicoesFamily sf
    INNER JOIN agrega a ON a.id_SubscicoesFamily = sf.id

```

```
INNER JOIN Contas f ON f.id = a.id_SubscicoesFree
)
ORDER BY id;
```

A pesquisa acima junta todas as pessoas que estejam associadas a SubscriçõesFamily (incluindo os agregados).

```
SELECT idMusica, nomeM
FROM Musicas
MINUS
SELECT m.idMusica, m.nomeM
FROM Musicas m
JOIN tem t ON m.idMusica = t.idMusica
MINUS
SELECT m.idMusica, m.nomeM
FROM Musicas m
JOIN contem c ON m.idMusica = c.idMusica;
```

A pesquisa acima devolve todos os singles (músicas que não pertencem a nenhum Album) que não estão em nenhuma Playlist, ou seja ninguém gosta delas.

## 6. Funcionalidades

### 6.1. Páginas de entrada

Na aplicação é possível aceder a outras páginas a partir de botões. Estes botões podem direcionar o utilizador para outras páginas ou para relatórios, o exemplo disto é a página inicial (“Home”) que tem diferentes opções que o utilizador pode escolher de acordo com o que necessita. Um bom exemplo é em “Contas” que podemos aceder a um relatório (“Todas as Contas”) ou a outras páginas.

### 6.2. Atributos de fácil compreensão

Em certos casos é vantajoso ter mais informação que o que está presente na tabela, como por exemplo em “Todas as subscrições” ou “Relatório artistas”. Em vez de deixarmos apenas na tabela o id da Conta decidimos colocar o seu nome e email para

ter uma maior facilidade na identificação da pessoa responsável pela Conta correspondente aquele id.

### 6.3. Valores derivados

Para verificar a popularidade dos Artistas decidimos implementar um relatório que mostra o número total de seguidores de cada Artista. Para aparecerem na tabela é necessário que tenham mais do que 1 seguidor, caso contrário não aparecem nesse relatório.

### 6.4. Atualização da base de dados

Em muitos dos relatórios é possível criar/adicionar, alterar ou remover conteúdos à base de dados. Para criar/adicionar é necessário carregar no botão “Create” no canto superior direito e adicionar a informação pedida. Em quase todos os casos é preciso ter atenção ao id inserido pois se não existir não será possível criar/adicionar. Para alterar informação temos de clicar no lápis no canto esquerdo e alteramos as informações que queremos. Atenção que nem sempre é possível executar este processo e por vezes a alteração tem de ser feita em relatórios exteriores. Para remover é necessário também poder alterar e clicamos no botão “Delete” no canto inferior direito. Tal como nas alterações nem sempre é possível remover podendo ter de ser feito em tabelas exteriores.

### 6.5. Lista de valores (LoV)

Em certos casos o utilizador pode não saber que tipo de opções existe ao seu dispor. Um exemplo disto é na adição de uma Subscrição Paga que pode ser Premium ou Family. Para facilitar decidimos adicionar as opções possíveis disponíveis para não causar confusões nem erros, facilitando, neste caso, a criação de uma nova Subscrição Paga. Também podemos encontrar esta funcionalidade em “Todas as Músicas” para selecionar um tipo de género de Música, aparecendo apenas os géneros existentes na aplicação.

### 6.6. Breadcrumbs

Quando estamos numa página temos a possibilidade de navegar até outra página que esteja diretamente ligada com essa. Esta funcionalidade está disponível em toda a nossa aplicação podendo aceder a esta funcionalidade clicando em uma das páginas presentes no canto superior esquerdo.

### 6.7. Relatórios interligados

Para melhorar a visualização da tabela que contem todas as Músicas decidimos organizar pelo género de Música. Ao alteramos o tipo de género de Música em

“Gêneros” com o LoV criado (6.5) irá aparecer o relatório com todas as Músicas desse gênero, assim tornando a pesquisa mais organizada e fácil.

## 6.8. Detalhe condicional

Depois de selecionarmos o gênero de Música em “Todas as Músicas” é nos dado um total de todas as Músicas desse gênero para aumentar a informação recebida pelo utilizador.

## 6.9. Master-detail

Utilizamos uma Master-detail na página das SubscriçõesPagas, com intenção de facilitar a gestão de Contas através da própria página. Permite-nos alterar, apagar e criar novas contas tudo a partir do Relatório de SubscriçõesPagas.

## 6.10. Consultas interessantes

A primeira escolha interessante é a visualização e criação de Subscrições Free. Visto que não temos acesso a todo o tipo de Subscrições, sejam Pagas do estilo Premium ou Family, também devemos ter um relatório para as Subscrições Free que não tenham todas as Subscrições lá, mas sim só que tem uma Subscrição Free e que não esteja num plano familiar de momento.

O número de seguidores ajuda a perceber a popularidade dos Artistas presentes. Eles são adicionados automaticamente a este relatório quando têm mais do que 1 seguidor.

Na “Subscrições Family” decidimos por colocar não só que pagou pela Subscrição Family mas também quem está presente do plano familiar pois assim termos uma melhor ideia de quem faz parte de uma Subscrição Family.

Por fim decidimos criar a possibilidade de visualizar quais das Músicas não tem sido popular para termos uma ideia de que tipo de Músicas não são muito ouvidas ou até de interessantes. As Músicas são adicionadas automaticamente quando não pertencem a nenhuma Playlist ou Álbum.

## 6.11. Início da subscrição paga

Sendo que apenas é possível adicionar um Pagamento associando-o a uma Subscrição Paga, decidimos que para que a Subscrição fique ativa só depois do Pagamento, a data de início da Subscrição fica NULL até ser efetuado um Pagamento válido. Ou seja, primeiro adicionamos as Subscrições Pagas e não irá ter data de início. Para alterar isso é necessário criar um Pagamento associado a essa Conta para ter uma data de início, simbolizando o início da Subscrição a partir do Pagamento da mesma. Caso seja feito update ao tipo da Subscrição, a data de início volta a não existir até que ocorra outro Pagamento.

## 7. Guia da Aplicação

### 7.1. Página inicial

- Contas: O utilizador é redirecionado para a página que gere Contas.
- Pagamentos: O utilizador é redirecionado para a página que gere os Pagamentos.
- Mídia: O utilizador é redirecionado para a página que gere tudo relacionado com Músicas.

### 7.2. Atributos de fácil compreensão

- Contas -> Subscrições -> “Qualquer botão (exceto Subscrições Pagas e Planos Familiares)”
- Contas -> Artistas -> Relatório artistas

Todos os relatórios referidos presentes em Subscrições e a dos Artistas têm informação detalhada sobre a Conta e não apenas o seu Id de Conta.

### 7.3. Valores derivados

- Contas -> Artistas -> Número de Seguidores

Este relatório apresenta o número total de seguidores de cada Artista (caso tenham).

### 7.4. Atualização da base de dados

- Contas -> Todas as Contas
- Contas -> Subscrições -> Subscrições Free
- Contas -> Subscrições -> Subscrições Pagas
- Contas -> Subscrições -> Planos Familiares
- Etc.

Nos relatórios acima mencionados é possível fazer alterações ao seu conteúdo.

## 7.5. Lista de valores (LoV)

- Contas -> Subscrições -> Subscrições Pagas -> Criar
- Mídia -> Todas as Músicas

No relatório de “Subscrições Pagas” podemos encontrar opções de escolha no tipo de Subscrição que queremos na criação de uma Subscrição Paga. Já em “Todas as Músicas” podemos escolher o tipo de género de Música que queremos.

## 7.6. Breadcrumbs

- Todas as páginas no canto superior esquerdo têm breadcrumbs que o direcionam para outra página.

## 7.7. Relatórios interligados

- Mídia -> Todas as Músicas

Nesta página é possível verificar vários relatórios dependendo do género de Música que escolhemos. Depois de escolher o género irá aparecer todas as Músicas desse género no relatório.

## 7.8. Detalhe condicional

- Mídia -> Todas as Músicas

Nesta página é possível ao escolher um género de Música e verificar quantas Músicas existem desse género específico na Base de Dados.

## 7.9. Master-detail

- Contas -> Subscrições -> Subscrições Pagas

Esta página é uma master-detail que possibilita a alteração de informação das Contas.

## 7.10. Consultas interessantes

- Contas -> Subscrições -> Subscrições Free: Apresenta todas as Contas com uma Subscrição Free, ou seja, que não tenham uma Subscrição Paga ou que estejam num grupo familiar.
- Contas -> Artistas -> Número de Seguidores: Dá-nos a informação de quantos seguidores tem cada Artista se tiverem pelo menos 1.
- Contas -> Subscrições -> Subscrições Family: Apresenta todas as Contas com uma Subscrição Family e todas as Contas que estejam num grupo familiar.

- Mídia -> Singles menos gostados: Todas as Músicas que não estão em uma Playlist nem em nenhum Álbum.

## 7.11. Início da subscrição paga

- Contas -> Subscrições -> Subscrições Pagas: Adicionamos uma Subscrição Paga que ira ficar sem data de início.

Posteriormente

- Pagamentos -> Relatório de Pagamentos: Adicionamos um pagamento associado à Subscrição Paga, com o valor correspondente ao Pacote escolhido.

Para atualizar ou remover

- Contas -> Subscrições -> Subscrições Pagas: Seleccionamos a Subscrição a ser alterada e após a alteração volta a ficar sem data de início (inativa), enquanto não houver novo Pagamento. Para remover é necessário que a Subscrição tenha expirado ou que não tenha data de início.