

Projeto de Base de Dados

Ano letivo 24/25

Apresentação do tema

Gestão da Plataforma Stremio

Versão - Servidor Local



Elementos do Grupo 17:

- Duarte Coelho, 65154
- Francisco Pereira, 67940
- Ricardo Batista, 68509

Turno Prático: P1

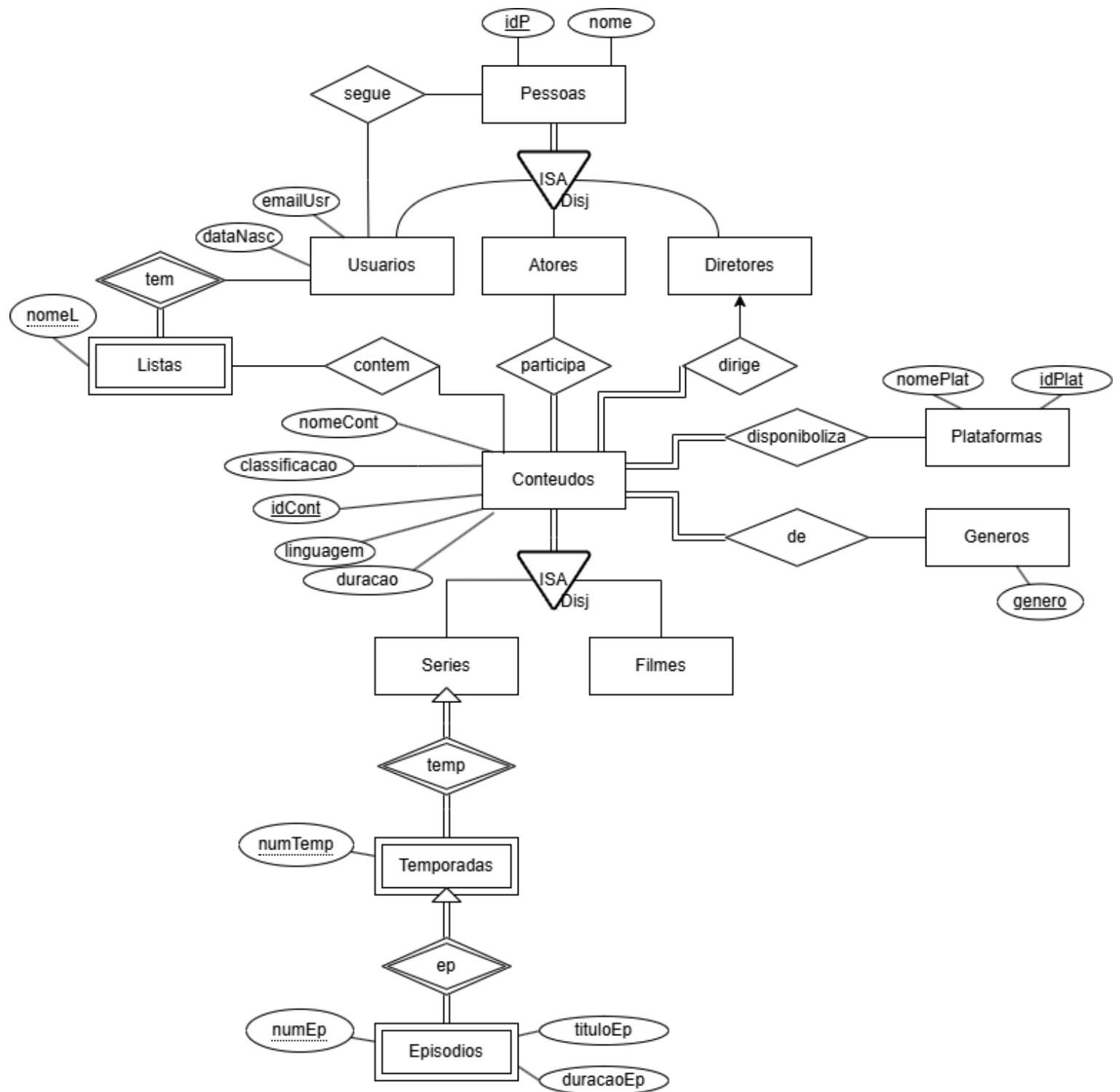
Descrição da Base de dados a Desenvolver

A plataforma Stremio pretende implementar uma base de dados para gerir informação sobre os diferentes tipos de conteúdos (filmes e séries), os seus utilizadores, atores e diretores.

Há assim que ter em conta que:

- A plataforma disponibiliza séries e filmes das mais conhecidas plataformas de streaming como a Netflix, Amazon Prime, HBO, entre outros. Para os filmes, queremos saber título, duração, linguagem, e um id único que o identifica.
- Para as séries queremos também ter informação sobre as temporadas, episódios e a sua duração total (soma da duração dos episódios). No caso dos episódios, terão a mesma informação que os filmes e ainda o número do episódio e o seu título. Para cada temporada temos apenas o número da temporada.
- Os conteúdos devem ter informação sobre a sua classificação, numa escala de 1 - 10, e sobre os seus vários géneros (terror, thriller, ação, etc..), devem também ter informação sobre o seu tipo (FILME ou SÉRIE).
- Para cada uma das plataformas de streaming queremos guardar o seu nome e um ID único que a identifica. Uma plataforma disponibiliza vários conteúdos, e conteúdos podem também ser disponibilizados por mais de uma plataforma.
- Queremos também ter informações sobre os vários usuários, isto inclui o seu nome, email, data de nascimento e id. Cada usuário tem várias listas de filmes e séries guardados por si, e também pode seguir outros usuários, atores e diretores, (sendo que um utilizador não pode seguir a si mesmo).
- Cada lista tem um nome, sabendo que utilizadores diferentes podem ter listas com nomes iguais, mas um utilizador não pode ter duas ou mais listas com o mesmo nome. Estas listas contêm os conteúdos desejados.
- Na plataforma temos também informação sobre os atores que participam em cada filme/série, e um único diretor por conteúdo. Para ambos queremos guardar um nome, e um identificador único.

Modelo ER



Modelo Relacional

Pessoas (idP, nome)

Usuarios (idU, emailUsr, dataNasc)

- idU é chave estrangeira de Pessoas(idP)

Atores (idA)

- idA é chave estrangeira de Pessoas(idP)

Diretores (idD)

- idD é chave estrangeira de Pessoas(idP)

Listas (nomeL, idU)

- idU é chave estrangeira de Usuarios

Conteudos (idCont, nomeCont, classificacao, linguagem, idD, duracao, tipo)

- idD é chave estrangeira de Diretores

Series (idCont)

- idCont é chave estrangeira de Conteudos

Filmes (idCont)

- idCont é chave estrangeira de Conteudos

Temporadas (idCont, numTemp)

- idCont é chave estrangeira de Series

Episodios (idCont, numTemp, numEp, tituloEp, duracaoEp)

- (idCont, numTemp) é chave estrangeira de Temporadas

Plataformas (idPlat, nomePlat)

Generos (genero)

segue(idU, idP)

- idU é chave estrangeira de Utilizadores(idP)
- idP é chave estrangeira de Pessoas (idP)

contem (idP, nomeL, idCont)

- (idP, nomeL) é chave estrangeira de Listas
- idCont é chave estrangeira de Conteudos

participa (idCont, idA)

- idCont é chave estrangeira de Conteudos
- idA é chave estrangeira de Atores

disponibiliza (idPlat, idCont)

- idPlat é chave estrangeira de Plataformas
- idCont é chave estrangeira de Conteudos

de (idCont, genero)

- idCont é chave estrangeira de Conteudos
- genero é chave estrangeira de Generos

Consultas Interessantes

- Consulta 1: Quais os atores que participaram em todos os filmes do Diretor “Christopher Nolan” (id = 10000000040)?

```
SELECT p.nome as "Nome Ator"
FROM Atores a
JOIN Pessoas p on a.idA = p.idP
WHERE NOT EXISTS (
    SELECT idCont
    FROM Conteudos natural join Filmes
    WHERE idD = 10000000040
    MINUS
    SELECT p.idCont
    FROM participa p
    WHERE p.idA = a.idA
);
```

- Consulta 2: Quais os diretores que já dirigiram um filme com uma classificação de pelo menos 4 estrelas?

```
SELECT DISTINCT p.nome AS "Nome Diretor"
FROM Filmes f
JOIN Conteudos c ON f.idCont = c.idCont
JOIN Diretores d ON c.idD = d.idD
JOIN Pessoas p ON d.idD = p.idP
WHERE c.classificacao >= 4;
```

- Consulta 3: Quais as séries, apresentando também o número de temporadas e a sua classificação, oferecidas pela plataforma “Netflix”?

```
SELECT c.nomeCont AS "Nome Serie", COUNT(t.numTemp) AS "Nº Temporadas",
c.classificacao AS "Classificação"
FROM Conteudos c
JOIN Series s ON c.idCont = s.idCont
JOIN Temporadas t ON s.idCont = t.idCont
JOIN disponibiliza d ON c.idCont = d.idCont
JOIN Plataformas p ON d.idPlat = p.idPlat
WHERE p.nomePlat = 'Netflix'
GROUP BY c.nomeCont, c.classificacao;
```

- Consulta 4: Listar todos os filmes de todas as plataformas, juntamente com o nome da Plataforma?

```
SELECT
  c.nomeCont AS "Titulo",
  c.classificacao AS "Classificação IMDB",
  c.linguagem as "Linguagem",
  p.nomePlat AS "Disponibilizado por"
FROM Filmes f
JOIN Conteudos c ON f.idCont = c.idCont
JOIN disponibiliza d ON f.idCont = d.idCont
JOIN Plataformas p ON d.idPlat = p.idPlat
order by "Titulo";
```

Triggers

- A inserção de um tuplo na tabela segue verifica que os IDs dos utilizadores são diferentes. Este trigger é necessário para proibir que um utilizador se siga a si próprio.

```
CREATE OR REPLACE TRIGGER prevent_self_follow
BEFORE INSERT ON segue
FOR EACH ROW
BEGIN
    IF :NEW.idU = :NEW.idP THEN
        RAISE_APPLICATION_ERROR(-20100, 'Users cannot follow themselves.');
```

- Garantia de que antes de inserir um tuplo na tabela de Usuários o seu ID não está já registado. Este trigger foi criado para que se tenha termos uma mensagem de erro específica e para reforçar o facto que o ISA de Pessoas para Usuários, Atores e Diretores é disjunto.

```
CREATE OR REPLACE TRIGGER trg_check_disjoint_Pessoas
BEFORE INSERT ON Usuarios
FOR EACH ROW
DECLARE
    v_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM Atores WHERE idA = :NEW.idU;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'A person cannot be both Usuario
and Ator.');
```

- Garantia de que antes de inserir um tuplo na tabela de Atores o seu ID não está já registado. Este trigger foi criado para que se tenha uma mensagem de erro específica e para reforçar o facto que o ISA de Pessoas para Usuários, Atores e Diretores é disjunto.

```
CREATE OR REPLACE TRIGGER trg_check_disjoint_Atores
BEFORE INSERT ON Atores
FOR EACH ROW
DECLARE
    v_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM Usuarios WHERE idU = :new.idA;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'A person cannot be both Ator and
        Usuario');
    END IF;
    SELECT count(*) into v_count from Diretores where idD = :new.idA;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR (-20100, 'A person cannot be both Ator and
        Diretor');
    END IF;
END;
/
```

- Garantia de que antes de inserir um tuplo na tabela de Diretores o seu ID não está já registado. Este trigger foi criado para que se tenha termos uma mensagem de erro específica e para reforçar o facto que o ISA de Pessoas para Usuários, Atores e Diretores é disjunto.

```
CREATE OR REPLACE TRIGGER trg_check_disjoint_Diretores
BEFORE INSERT ON Diretores
FOR EACH ROW
DECLARE
    v_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM Usuarios WHERE idU = :new.idD;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'A person cannot be both Diretor
        and Usuario');
    END IF;
    SELECT count(*) into v_count from Atores where idA = :new.idD;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR (-20100, 'A person cannot be both Diretor
        and Ator');
    END IF;
END;
/
```


- Antes da inserção de um Filme verificação de que o ID não está registado na tabela Series. Este trigger foi criado para apresentar uma mensagem de erro específica e para reforçar o facto que o ISA de Conteúdos para Filmes e Séries é disjunto.

```
CREATE OR REPLACE TRIGGER trg_check_disjoint_Filmes
BEFORE INSERT ON Filmes
FOR EACH ROW
DECLARE
    v_exists NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_exists
    FROM Series
    WHERE idCont = :NEW.idCont;

    IF v_exists > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'This content ID is already
registered as a Series.');
```

Descrição: Trigger que antes de inserir uma Série verifica que o ID não está registado na tabela Filmes. Criamos este trigger para termos uma mensagem de erro específica e para reforçar o facto que o ISA de Conteúdos para Filmes e Séries é disjunto.

```
CREATE OR REPLACE TRIGGER trg_check_disjoint_series
BEFORE INSERT ON Series
FOR EACH ROW
DECLARE
    v_exists NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_exists
    FROM Filmes
    WHERE idCont = :NEW.idCont;
    IF v_exists > 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'This content ID is already
registered as a Film.');
```

Descrição: Trigger que verifica que a classificação de um conteúdo não pode ser menos que 0.0 ou mais de 10.0. Criamos este trigger para reforçar o facto que o valor de uma classificação não pode ser menos de 0.0 ou mais de 10.0.

```
CREATE OR REPLACE TRIGGER trg_check_classificacao
```

```

BEFORE INSERT OR UPDATE ON Conteudos
FOR EACH ROW
BEGIN
    IF :NEW.classificacao < 0.0 OR :NEW.classificacao > 10.0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'Classificacao must be between 0.0
and 10.0.');
```

Descrição: Trigger que automaticamente regista um Conteúdo como Filme ou Série dependendo do valor do atributo tipo no tuplo inserido em Conteúdos. Criamos este trigger para ajudar na inserção de dados na BD.

```

CREATE OR REPLACE TRIGGER trg_after_insert_conteudos
AFTER INSERT ON Conteudos
FOR EACH ROW
BEGIN
    IF :NEW.tipo = 'SERIE' THEN
        INSERT INTO Series (idCont) VALUES (:NEW.idCont);
    ELSIF :NEW.tipo = 'FILME' THEN
        INSERT INTO Filmes (idCont) VALUES (:NEW.idCont);
    END IF;
END;
/
```

Descrição: Trigger que automaticamente atualiza o valor de duração para a série em que foi adicionado um episódio. Criamos este trigger para garantir que o valor de duração na tabela Conteúdos, referente a Séries, está sempre atualizado.

```

CREATE OR REPLACE TRIGGER trg_update_duracao_on_update
AFTER UPDATE ON Episodios
FOR EACH ROW
BEGIN
    UPDATE Conteudos
    SET duracao = (
        SELECT SUM(duracaoEp)
        FROM Episodios
        WHERE idCont = :NEW.idCont
    )
    WHERE idCont = :NEW.idCont;
END;
```

Limitações/Opções

- Uma das opções tomadas ao fazer esta base de dados foi que só há um diretor por conteúdo de modo a simplificar a sua dimensão.
 - Na tabela dos conteúdos tivemos também de inserir um atributo extra que é o tipo, para ajudar com a inserção através do APEX e do o SQL. Acrescentamos também a duração nesta tabela, que anteriormente estava só nos filmes e episódios, para simplificar a inserção no APEX.
-

Alterações da primeira versão do Projeto

Em primeiro lugar, foi removida a entidade dos métodos de pagamento, principalmente porque o nosso projeto é baseado na plataforma Stremio que, na realidade, não tem essa opção.

A relação “avalia” foi removida uma vez que no Stremio não é assim que funciona e a classificação é apenas apresentada de fontes exteriores. Também não faria sentido o utilizador ter a capacidade de avaliar algo que nunca tenha visto, o que faz com que a remoção desta relação simplifique o código, evitando a implementação desta verificação. Foi adicionado um atributo “Classificação” na tabela “Conteúdos” para substituir esta relação.

Criamos a entidade “Pessoas” para generalizar usuários, atores e diretores. Tornou também possível, a criação da relação segue que achamos que se integrava no nosso trabalho.

Trocamos a relação “Lista” por uma entidade com o mesmo nome. Isto pois, anteriormente os conteúdos só podiam ser uma vez relacionados com o utilizador, ou seja um utilizador não podia ter o mesmo conteúdo em listas diferentes. Devido a esta nova entidade, adicionamos duas novas relações, uma das “Listas” para os “Conteúdos” e outra das “Listas” para os “Usuários”, para poder interligar tudo.

Como referimos anteriormente a tabela “Conteúdos” e “Filmes” foram alteradas, tendo ganho dois novos atributos(duração e o tipo) e perdido um(duração) respetivamente.

Por último, alteramos a idade por data de nascimento no que diz respeito a entidade dos “Usuários” e retiramos a sinopse de “Conteúdos” e “Episódios” devido à dificuldade de inserção de dados desse tipo.

Interface

A nossa interface é constituída por várias páginas, sendo cada uma responsável por apresentar os dados da nossa base de dados em lista. Temos também forms para inserir novos dados, de forma mais simples para o utilizador.

Manual do Utilizador

Página Inicial

Na página inicial temos vários botões que nos levam para as diferentes páginas do programa, cada um com um nome correspondente a sua página.

Página Series e Filmes

Nestas páginas simplesmente listamos todas as Séries e Filmes da base de dados respetivamente, com o seu título, classificação IMDB, linguagem, e a plataforma que a disponibiliza. O utilizador pode através do botão “Actions” filtrar e formatar a tabela da maneira preferida, de modo a mostrar os dados desejados.

Página Utilizadores

Nesta página listamos todos os Utilizadores registados na aplicação, dando informação sobre o seu nome, email, data de nascimento e idade, nesta página um utilizador pode clicar num dos nomes dos utilizadores e ser redirecionado para uma página com as Listas do utilizador selecionado, mostrando o nome de cada Lista e o número de conteúdos dentro da mesma.

Nessa página o utilizador pode depois clicar no nome de uma das Listas e ser outra vez redirecionado para uma página com todos os conteúdos dentro dessa Lista, mostrando o nome do conteúdo, se é uma série ou filme, a sua classificação, e a sua linguagem.

Página Temporadas

Nesta página temos um form master detail, sendo a tabela Temporadas o Master e a tabela Episódios o detail. Aqui o utilizador pode ver todas as temporadas registadas na base de dados, e editar os episódios de cada uma.

Páginas das Consultas Interessantes

Nestas quatro páginas (Consulta 1, Consulta 2, Consulta 3, e Consulta 4) mostramos os resultados de cada uma das nossas consultas interessantes definidas acima. O utilizador pode utilizar o botão “Actions” para adaptar os dados mostrados a seu gosto.

Página Relatório Conteúdos

Esta página lista todos os conteúdos disponíveis na aplicação, mostrando para cada um o seu nome, classificação, idioma, e o seu diretor. Aqui o utilizador pode clicar no botão “Create” e ser redirecionado para um form onde pode selecionar um tipo de conteúdo que queira inserir, adicionar os dados necessários e por fim adicionar o novo conteúdo a base de dados.

Neste form, temos duas select lists, para facilitar a definição do tipo de conteúdo, e do seu diretor, e temos também uma funcionalidade que ao selecionar o tipo "Série" automaticamente o valor duração fica a 0 (não podendo ser alterado), pois ao criar a série série não existe ainda nenhum episódio, e logo a soma das durações de cada episódio é 0.

Página Relatório Plataformas

Esta página é um simples form com report, que apresenta as Plataformas já registradas na plataforma e permite ao utilizador criar novas Plataformas, através do botão "Create".

Página Relatório Temporadas

Esta página é um simples form com report, que apresenta as Séries registradas e o número de temporadas que cada uma tem na plataforma e permite ao utilizador criar novas Temporadas, através do botão "Create".

Página Relatório Episódios

Esta página é um simples form com report, que apresenta os episódios registrados na plataforma e permite ao utilizador criar novos episódios, através do botão "Create".

Página Relatório Disponibiliza

Esta página é um simples form com report, que apresenta os nomes dos conteúdos e as plataformas que os disponibilizam registrados na aplicação e permite ao utilizador criar novas relações que dizem que plataforma disponibiliza que conteúdo, através do botão "Create".

Página Relatório relação De

Esta página é um simples form com report, que apresenta os nomes dos conteúdos e os géneros de cada um registados na aplicação e permite ao utilizador criar novas relações que dizem que género/s pertence a cada conteúdo, através do botão "Create".

Implementações no Apex

- Existência de uma página de entrada onde existam ligações para as páginas subsequentes
 - Página Inicial, implementada com o uso de botões que redirecionam para as várias páginas da Interface.
- Listagem de dados da Base de Dados onde códigos referentes a chaves externas sejam substituídos pelo valor de outros atributos de fácil compreensão
 - Pode ser visto em quase todas as páginas da Interface, como por exemplo na página do Filmes onde metemos o nome da Plataforma que disponibiliza em vez do seu identificador.
- Listagem de dados da Base de Dados onde sejam apresentados valores derivados (e.g. a média de cada aluno)
 - Na página dos Utilizadores temos a sua idade, sendo esta derivada da data de nascimento do utilizador.
- Possibilidade de inserir, remover e atualizar tuplos da Base de Dados
 - Em qualquer um dos relatórios podemos inserir novos tuplos na base de dados, e na página Temporadas podemos remover e alterar os dados dos Episódios registrados.
- Possibilidade de preencher valores de atributos correspondentes a relações (chaves externas) sem se ter conhecimento de códigos
 - Em qualquer form na base de dados onde se possa inserir dados referentes a chaves externas temos lista de valores para serem selecionados, por exemplo na inserção de um conteúdo podendo escolher o nome do Diretor sem saber o seu identificador.
- Existência de dois reports interligados, onde um apresenta detalhes do outro (drill-down)
 - Na página dos utilizadores é possível clicar um utilizador e ver as suas Listas, e nessa página é também possível clicar na lista e ver os seus Conteúdos.
- Existência de um form master-detail, onde seja possível inserir, remover e alterar dados (pelo menos) do detail
 - Temos a página de Temporadas, que é um form master detail que permite inserir, alterar, e remover os dados do detail (Episódios).