

**BASE DE DADOS**

**Plataforma de Gestão de um *Private Torrent Tracker***

**Trabalho Prático - P4G7

**MESTRADO INTEGRADO EM ENGENHARIA COMPUTACIONAL**

Álvaro Freixo, NMEC-93116

João Maria Machado, NMEC-89132

# **ÍNDICE**

[**ÍNDICE** 2](#_Toc74931103)

[**INTRODUÇÃO** 3](#_Toc74931104)

[**MODELO DE DADOS** 4](#_Toc74931105)

[**Analise de Requisitos** 4](#_Toc74931106)

[**Diagrama de Entidade Relação** 5](#_Toc74931107)

[**Esquema Relacional** 6](#_Toc74931108)

[**DATA DEFINITION LANGUAGE (DDL)** 7](#_Toc74931109)

# **INTRODUÇÃO**

Neste trabalho pretende desenvolver-se uma plataforma de distribuição de *torrents* – um *torrent tracker* (meramente para fins educacionais) como forma de aplicar na prática os conceitos lecionados na unidade curricular de base de dados. Na conceção de uma base de dados, num contexto, real é importante conseguir relacionar dados completamente diferentes provenientes de realidades fundamentalmente diferentes. Por isso, o nosso grupo optou por este tema, numa plataforma de *torrent tracking* é possível relacionar dados de “locais” completamente opostos da base de dados o que obriga à utilização de *querys* complexas e implementação cuidadosa de *stored procedures* na base de dados e também obriga em vários casos à manipulação de dados de modo a conseguir derivar atributos importantes na gestão da plataforma.

A base de dados concebida foi baseada numa plataforma de *torrente tracking* real. A plataforma em questão trata-se de uma plataforma privada com uma comunidade coesa e fechada com regras próprias. As regras são explicitadas no documento de requisitos elaborado pelo grupo (o presente relatório subentende a leitura prévia do documento de requisitos). No entanto, as regras podem ser sumarizadas no seguinte princípio: “As regras em causa, existem para garantir a sobrevivência do conteúdo e a satisfação da comunidade “. O modelo de dados teve, evidentemente de ser ajustado, para permitir o policiamento das regras impostas.

O presente relatório encontra-se divido em 3 partes. Uma primeira parte na qual é explicada o modelo de dados dando destaques a algumas tabelas centrais ao projeto, uma segunda parte focada na introdução das tabelas e tuplos na base de dados e finalmente, uma terceira parte focada nos testes feitos à integridade do modelo e a sua validade.

Para além da análise de requisitos a leitura deste relatório pressupõem que leitor tem acesso ao diagrama de entidade relação, ao esquema relacional, ao script que contém o DDL (*data definition language*)*,* ao script que contém o DML (*data manipulation language*), ao *script* que contém os tuplos que foram inseridos para testar a base de dados e finalmente, para a compreensão do relatório também é necessário que leitor tenha acesso ao GUI desenvolvido pelo grupo, que por sua vez implementa de maneira prática o trabalho desenvolvido.

# **MODELO DE DADOS**

## **Analise de Requisitos**

Antes de proceder à criação da base de dados propriamente dita foi necessário discutir quais os requisitos prévios à alocação dos dados. Para tal efeito, procedeu-se à elaboração de uma **lista de requisitos.** Esta lista resulta de uma cuidada análise, discutida em grupo. Esta lista teve por base um site real, como tal com base no site analisado, começamos por enumerar e diferenciar tipos de dados necessários à criação de um diagrama de entidade\relação. Um dos aspetos a sublinhar no processo é a criação de uma estrutura de dados capaz de suportar vários tipos de utilizadores com papeis diferentes na comunidade, nomeadamente: um utilizador norma capaz de descarregar conteúdo, um utilizador conhecido como ***uploader*** responsável pela inserção de novos conteúdos na plataforma, um utilizador do tipo ***staff*** capaz de moderar as interações na plataforma e recompensar utilizadores pelo bom comportamento e finalmente um utilizador do tipo ***admin*** com plenos poderes sobre a plataforma.

Um outro aspeto que importa relevar é a criação de uma estrutura interna de organização de conteúdos, isto é, para melhorar a organização da plataforma e oferecer ao utilizador uma experiência de pesquisa mais amigável dividimos os conteúdos em 3 categorias: Programas, Jogos, Conteúdo cuja informação é detalhada num site de critica de cinema (que por sua vez se subdivide em séries e filmes).

Como forma de permitir interação entre membros da comunidade e conteúdo implementou-se um sistema de comentários de modo a que todos os utilizadores possam tecer comentários relativos a qualquer conteúdo disponibilizado pela plataforma.

Finalmente, para incentivar o crescimento da comunidade, implementou-se um sistema de prémios segundo o qual, utilizadores podem ser recompensados pela boa conduta de interação com conteúdos ou até mesmo por recrutarem outros utilizadores

Com todos estes requisitos considerados, procedeu-se à elaboração de um **diagrama de entidade\relação**.

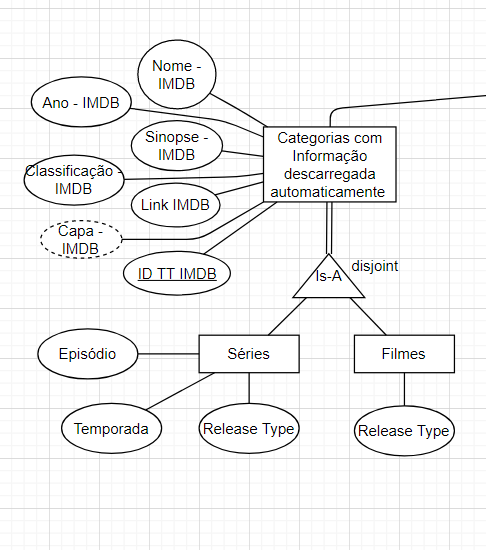
**Diagrama de Entidade Relação**

O diagrama de entidade relação consiste numa planificação da maneira de como a informação será organizada na base de dados, ou seja, após todas as considerações feitas pela análise de requisitos é possível começar a planificar a estrutura dos dados bem como a maneira como estes se relacionam entre si.

Um processo, importante, que acompanha a criação deste diagrama corresponde à escolha de chaves primárias para cada entidade incorporada no próprio diagrama. Uma chave primária é uma chave que identifica de maneira única um tuplo numa tabela. A atribuição de uma chave primária deve ser efetuada cuidadosamente, pois esta tem de ser imperativamente imutável, ou seja, esta chave não pode consistir numa caraterística que possa ser mudada com interações com os dados, caso contrário a integridade dos dados pode ser seriamente comprometida.

Um dos casos que mais importa mencionar é o caso do utilizador, no qual foi necessário criar-se um atributo próprio para atuar como chave primária – **ID\_User**, é de notar que este atributo foi definido como um ***integer* (*INT*)** que é automaticamente incrementado à medida que os utilizadores vão aderindo à plataforma. Esta estratégia de escolha de chaves primárias foi utilizada em várias tabelas, desse modo garantindo a integridade dos dados na BD.

Uma outra situação que se revelou importante a quando da criação do diagrama de entidade relação (e consequentemente durante todo o processo de criação da base de dados) foi o uso adequado de ***disjoints*** (***IS A***). Um *disjoint* consiste numa classificação de dados que afirma que os dados classificados pertencem necessariamente a uma (ou várias) entidades posteriormente especificadas. Um exemplo desta implementação no nosso modelo de dados pode ser observado na figura 2.1.



.

**Figura 2.1**: Exemplo de um *disjoint* na implementação do diagram entidade\relação. Neste caso podemos observar que uma categoria cuja informação é descarregada automaticamente é uma série ou filme.

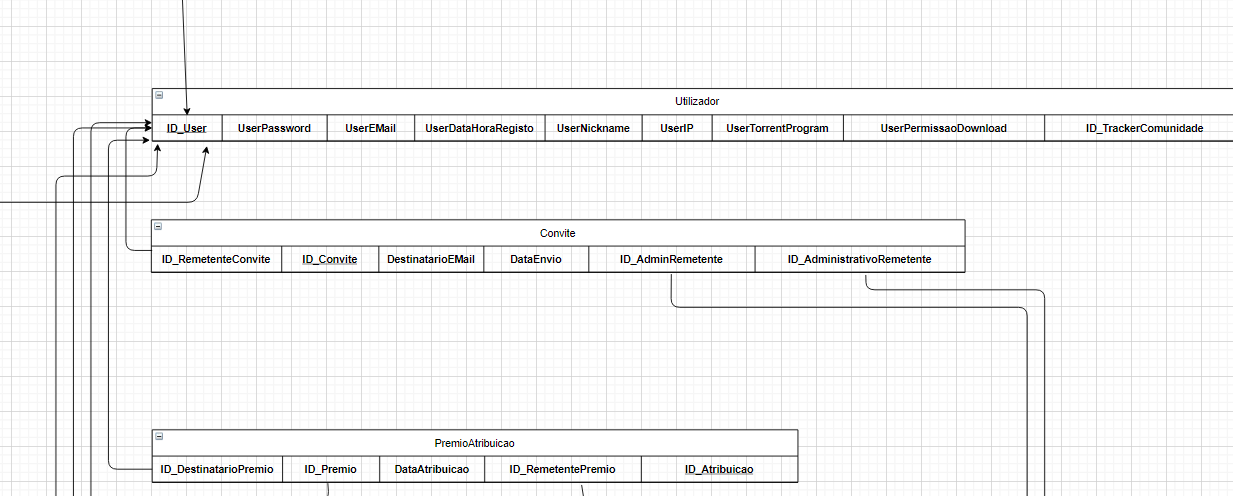
Finalmente após levar em consideração as propriedades previamente discutidas, foi possível elaborar um esquema de entidade\relação neste esquema estão explicitados todos os dados que possam , mesmo que remotamente, ser uteis na construção da base de dados de um *private torrente tracker* bem como a maneira como: estes se relacionam entre si (relações), a cardinalidade dos diferentes domínios dos atributos, explicitação das chaves primárias e atributos derivados, e a também a obrigatoriedade que cada entidade tem (ou não) em participar numa relação.

**Esquema Relacional**

Após a criação do diagrama de entidade\relação pode agora proceder-se à criação do **esquema relacional**. Contrariamente ao diagrama de entidade relação, o esquema relacional consiste numa forma de visualizar os dados sob a forma de tabelas, como maneira de diminuir o nível de abstração no processo de implementação e permitir a aproximação à linguagem SQL. Ou seja, a passagem do DER (diagrama entidade\relação) para o ER (esquema relacional) implica uma aproximação entre aquilo que foi previamente discutido a nível de requisito e a implementação que será feita no DDL (*data definition language*).

No processo de transição do DER para o ER, aquilo que era previamente definido como atributo (ou até mesmo relação) será agora sob a forma de uma tabela na qual são explicitados os seus atributos (também evidenciando a chave primária).

Nesta etapa ocorre também um processo extremamente importante no âmbito da questão da integridade dos dados, a definição de chaves estrangeiras. Uma chave estrangeira (**Foreign key**) consiste numa chave que é importada de outra tabela. A omissão, ou definição indevida de chaves estrangeiras pode facilmente levar a conflitos de dados. A figura 2.2 mostra um exemplo da definição de chaves estrangeiras.



**Figura 2.2**: Exemplo de uma tabela cuja chave (*ID\_User*) é chave estrangeira de 8 tabelas diferentes, o que é expectável, visto várias entidades são moldadas pelo utilizador.

Uma última etapa a considerar antes de prosseguir para a DDL é a normalização de dados. Este processo tem apenas um propósito: a eliminação de redundâncias. Para tal efeito identificam-se potências dependências funcionais nas tabelas do ER, isto é, maneiras como dados numa tabela se relacionam entre si. Seguidamente classificam-se as dependências funcionais. Através da eliminação ordenada e sucessiva dos diferentes tipos de dependências funcionais podemos a atingir a forma normal de Boyce-Codd (BCNF). Este algoritmo foi aplicado\verificado em todas as tabelas do nosso ER, desse modo evitando potenciais redundâncias.

# **DATA DEFINITION LANGUAGE (DDL)**

## **Criação de Tabelas**

Nesta secção do relatório será explorado o processo de implementação do modelo de dados em SQL. Serão abordadas alguns dos princípios que, durante a realização do trabalho, se mostraram determinantes.

 Na secção anterior foi elaborado um esquema relacional, que nesta etapa será o principal guia para a criação de tabelas no SQL. O trecho de código 3.1 exemplifica a criação da tabela de utilizador.

**Figura 2.3**: Exemplo de criação de uma tabela em SQL.