

Faculdade de Engenharia da Universidade do Porto



**Estudantes & Autores:**

André Leonor [up201806860@fe.up.pt](mailto:up201806860@fe.up.pt)

Carlos Veríssimo [up201907716@fe.up.pt](mailto:up201907716@fe.up.pt)

João Vasconcelos [up201504397@fe.up.pt](mailto:up201504397@fe.up.pt)

Grupo 601

Trabalho realizado no âmbito da unidade curricular de Bases de Dados  
([FEUP - Bases de Dados](#))

## A. Definição do Modelo Conceptual

O projeto irá representar uma base de dados de uma plataforma de partilha e visualização de vídeos. De cada pessoa ter-se-á a informação relativa ao IP e ao dispositivo a ser utilizado, sendo que um utilizador pode possuir um canal, ou não estar registado. Pessoas não registadas apenas podem assistir a vídeos, e um canal (pessoa identificada) pode subscrever outros canais (subscrição normal ou premium). Cada vídeo é criado por um canal, sendo que a data de publicação será sempre igual, ou mais tardia, que a data de carregamento do vídeo. Os canais podem interagir com os vídeos através da atribuição de classificações, atribuídas pelos vários canais a diversos vídeos, partilha, em que os canais podem partilhar os vários vídeos, ou comentários, em que um determinado por um canal, e, por sua vez, um canal pode atribuir um dislike ou um like aos comentários dos vídeos. Um utilizador apesar de poder reagir a vários vídeos e comentários, não pode ter duas reações diferentes em simultâneo (não pode dar like e dislike no mesmo vídeo ou comentário).

Uma lista de reprodução é criada por um canal, sendo que um canal pode criar várias listas de reprodução, em conjuntos de vídeos, sendo, portanto, a duração da lista de reprodução correspondente à soma das durações dos vídeos que constituem a lista.

Os vídeos possuem alguns atributos não estáticos, sendo eles os números de visualizações, likes, dislikes, comentários e partilhas e com base no histórico de visualizações dos utilizadores, torna-se possível formular uma lista dos 50 vídeos mais populares (tendências) para cada país.

O tempo mínimo de uma visualização são 10 segundos. e os seguintes atributos são únicos:

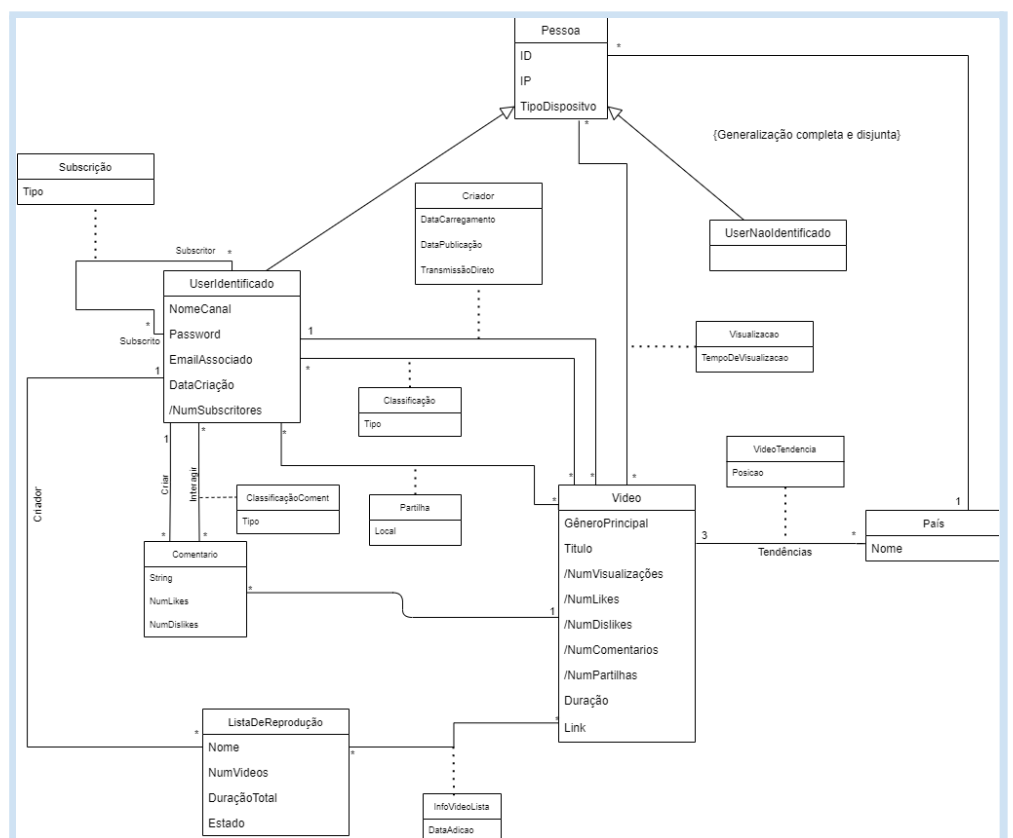
- o Atributo **NomeCanal** da tabela **USERIDENTIFICADO**;
- o Atributo **EmailAssociado** da tabela **USERIDENTIFICADO**;
- o Atributo **Link** da tabela **VIDEO**;
- o Atributos do tipo **Id\_ "tabela"** usados na criação da base de dados. Ex: **LogUser\_Id**, **Id\_Pais**, **Id\_Video**, entre outros...

Uma lista de reprodução terá um atributo estado, cujos valores poderão apenas ser: 'Pública' ou 'Privada'

Da mesma forma, a classe de associação "Criador", entre UserIdentificado e Vídeo, tem um atributo chamado TransmissãoDireto cujos valores válidos serão apenas 'Sim' ou 'Não'.

A data de publicação do vídeo será, obrigatoriamente maior ou igual à data de carregamento do vídeo na plataforma.

Diagrama original e atualizado em: [Diagrama UML](#)



## B. Definição do esquema relacional

*/\* Object-oriented usado na generalização \*/*

Pessoa (Id\_Pessoa, Ip, TipoDispositivo, Id\_pais->Pais)

UserIdentificado (LogUser\_Id->Pessoa, Ip, TipoDispositivo, NomeCanal, Password, EmailAssociado, DataCriacao, NumSubscritores)

UserNaoidentificado(NotLogUser\_Id->Pessoa, Ip, TipoDispositivo)

Pais (Id\_Pais, Nome)

Video (Id\_Video, Link, Titulo, GeneroPrincipal, NumVisualizacoes, NumDislikes, NumLikes, NumComentarios, NumPartilhas, Duracao)

Comentario( Id\_Coment, String, NumLikes, NumDislikes, video->Video, userIdentificado->UserIdentificado)

ListaDeReprodução ( Id\_Lista, Nome, NumVideos, DuracaoTotal, Estado, userIdentificado->UserIdentificado)

Partilha (userIdentificado->UserIdentificado, video->Video, Local)

Classificacao(userIdentificado->UserIdentificado, video->Video, Tipo)

Visualizacao (peessoa->Pessoa, video->Video, TempoDeVisualizacao)

InfoVideoLista (listaDeReproducao->ListaDeReproducao, video->Video, DataAdicao)

VídeoTendencia (video->Video, pais->Pais, Posicao)

Subscricao (userIdentificado-> UserIdentificado, userIdentificado->UserIdentificado, Tipo)

ClassificacaoComent(userIdentificado-> UserIdentificado, comentario->Comentario, Tipo)

Criador (DataCarregamento, DataPublicacao, TransmissaoDireto, video->Video, userIdentificado->UserIdentificado)

## C. Análise de dependências funcionais e formas normais

Pessoa:

{Id\_Pessoa} -> {Ip, TipoDispositivo, Id\_Pais}

UserIdentificado :

{LogUser\_Id} -> {Ip, TipoDispositivo, NomeCanal, Password, EmailAssociado, DataCriacao, NumSubscritores}

UserNaoidentificado:

{NotLogUser\_Id} -> {Ip, TipoDispositivo}

Pais:

{Id\_Pais} -> {Nome}

{Nome} -> {Id\_Pais}

Video:

{Id\_Video} -> {Link, Titulo, GeneroPrincipal, NumVisualizacoes, NumDislikes, NumLikes, NumComentarios, NumPartilhas, Duracao}

{Link} -> {Id\_Video, Titulo, GeneroPrincipal, NumVisualizacoes, NumDislikes, NumLikes, NumComentarios, NumPartilhas, Duracao}

Comentario:

{Id\_Coment} -> {String, NumLikes, NumDislikes, video, userIdentificado}

ListaDeReprodução:

{Id\_Lista} -> {Nome, NumVideos, DuraçãoTotal, Estado, userIdentificado}

Partilha:

{userIdentificado, Video} -> {Local}

Classificação:

{userIdentificado, Video} -> {Tipo}

Visualização:

{Pessoa, Video} -> {Tempo De Visualizacao}

InfoVideoLista:

{listaDeReprodução, vídeo} -> {DataAdicao}

VídeoTendencia:

Não existem dependências funcionais não triviais.

Subscrição:

{UserIdentificado, UserIdentificado} -> {Tipo}

ClassificaçãoComent:

{UserIdentificado, Comentario} -> {Tipo}

Criador:

{Video} -> {DataCarregamento, DataPublicacao, TransmissaoDireto, UserIdentificado}

Uma relação R está conforme a Boyce-Codd Normal Form (BCNF) se em todas as dependências funcionais não triviais  $A \rightarrow B$ , A seja uma superkey, isto é, A + contenha todos os atributos de R. Sendo que cada elemento da esquerda é uma superkey, e, os atributos da direita dependem dessa mesma superkey, isto é, a partir de todas as dependências funcionais identificadas foi possível determinar todos os atributos das tabelas. Logo não ocorrem violações à BCNF, não ocorrendo também violações à 3ªFN, uma vez que 3ªFN é um subset de BCNF.

## D/E. Criação e adição de restrições à base de dados em Sqlite

- Criação das tabelas (classes) a ser incorporadas na base de dados:

```
-- Tabela referente a classe: PESSOA
DROP TABLE IF EXISTS PESSOA;
CREATE TABLE PESSOA ( ...
);
```

- Criação das colunas de cada tabela (atributos) e especificação do tipo de dados correspondente

```
-- Tabela referente a classe: PESSOA
DROP TABLE IF EXISTS PESSOA;
CREATE TABLE PESSOA (
    Id_pessoa INTEGER;
    Ip TEXT;
    TipoDispositivo TEXT;
    Id_Pais INTEGER;
```

- Introdução de restrições relativamente às colunas (atributos) da tabela criada

```
-- Tabela referente a classe: PESSOA
DROP TABLE IF EXISTS PESSOA;
CREATE TABLE PESSOA (
    Id_pessoa INTEGER NOT NULL UNIQUE,
    Ip TEXT DEFAULT 'NULL',
    TipoDispositivo TEXT DEFAULT 'NULL',
    Id_Pais INTEGER,
```

- Especificação das chaves primárias e chaves estrangeiras, assim como a especificação do procedimento quando se apaga/atualiza os valores:

```
-- Tabela referente a classe: PESSOA
DROP TABLE IF EXISTS PESSOA;
CREATE TABLE PESSOA (
    Id_pessoa INTEGER NOT NULL UNIQUE,
    Ip TEXT DEFAULT 'NULL',
    TipoDispositivo TEXT DEFAULT 'NULL',
    Id_Pais INTEGER,

    CHECK (Id_pessoa > 0),

    CONSTRAINT PESSOA_PK PRIMARY KEY(Id_pessoa),
    CONSTRAINT PESSOA_PAIS_FK FOREIGN KEY(Id_Pais) REFERENCES PAIS(Id_Pais) ON DELETE SET NULL ON UPDATE CASCADE
);
```

- Introdução de outras restrições relativamente às colunas (atributos) da tabela criada

```
    Password TEXT NOT NULL CHECK(LENGTH(Password) > 8),
    Estado TEXT NOT NULL CHECK(Estado = "Pública" OR Estado = "Privada"),
    Duração INTEGER NOT NULL CHECK (Duração > 0),
    CONSTRAINT DATA_VALID CHECK (DataCarregamento <= DataPublicacao),
```

Entre outras....

## F. Carregamento de dados

- Nesta parte do projeto, foi criado um ficheiro **povoar.sql**, ficheiro este que contém todos os comandos necessários para introduzir dados na base.

```
INSERT INTO PAIS VALUES (1, 'Portugal');
INSERT INTO PAIS VALUES (2, 'United States');
INSERT INTO PAIS VALUES (3, 'Brazil');
INSERT INTO PAIS VALUES (4, 'United Kingdom');
INSERT INTO PAIS VALUES (5, 'Spain');
INSERT INTO PAIS VALUES (6, 'France');
INSERT INTO PAIS VALUES (7, 'Australia');
INSERT INTO PAIS VALUES (8, 'Italy');
INSERT INTO PAIS VALUES (9, 'Germany');
INSERT INTO PAIS VALUES (10, 'Canada');
```

- Nesta parte, é também verificada a integridade das restrições especificadas na criação da base de dados. Abaixo encontram-se algumas dessas restrições a impedir a inserção de dados inválidos na base.

- Inserção de valores com o mesmo UNIQUE Id;

```
sqlite> INSERT INTO PAIS VALUES (1, 'Portugal');
sqlite> INSERT INTO PAIS VALUES (1, 'United States');
Error: UNIQUE constraint failed: PAIS.Id_Pais
```

- Inserção de valores que não obedecem o CHECK especificado;

```
CONSTRAINT DATA_VALID CHECK (DataCarregamento <= DataPublicacao),
sqlite> INSERT INTO CRIADOR(DataCarregamento, DataPublicacao, TransmissaoDireto, Id_Video, LogUser_Id)
VALUES("2021-02-16", "2019-02-16", 'Sim', 7, 4);
Error: CHECK constraint failed: DATA_VALID
```

- Inserção de valores que não constam no CHECK que especifica os valores válidos;

```
sqlite> INSERT INTO CLASSIFICACAO(LogUser_Id, Id_Video, Tipo)
...> VALUES(3, 2, 'DISL');
Error: CHECK constraint failed: Tipo = 'LIKE' OR Tipo = 'DISLIKE'
```

- Inserção de valores cujo tamanho se encontra fora daqueles aceites pela database;

```
sqlite> INSERT INTO USERIDENTIFICADO(Ip, TipoDispositivo, NomeCanal, Password, EmailAs
VALUES('7756.5559.7759.2999', 'mobile', 'Another interests', 'ajsf', '
Error: CHECK constraint failed: LENGTH>Password) > 8
```

- Entre outros...

- Verifica-se que a inserção de dados na base ocorreu sem qualquer tipo de problema através de uma GUI(*graphical user interface*) ou mesmo através da consola de um *IDE*.

Id_Pais	Nome
1	Portugal
2	United States
3	Brazil
4	United Kingdoom
5	Spain
6	France
7	Australia
8	Italy
9	Germany
10	Canada

```
sqlite> SELECT * from PAIS;
1|Portugal
2|United States
3|Brazil
4|United Kingdoom
5|Spain
6|France
7|Australia
8|Italy
9|Germany
10|Canada
```

Id_Coment	String	NumLikes	NumDislikes	Id_Video	LogUser_Id
1	Don't forget to subscribe!	1	1	1	1
2	Nice video	0	0	1	2
3	Cringe	0	0	3	4
4	Interesting video	0	0	3	3
5	very good	0	0	7	2
6	another one	1	1	5	1
7	ahahahahahah	3	1	5	1
8	great match!!!!	4	0	7	1
9	Bro, you are crazy!	1	0	8	8
10	NULL	0	0	12	3
11	Poor Ethan	1	0	13	1

```
sqlite> SELECT * from COMENTARIO;
1|Don't forget to subscribe!|1|1|1|1
2|Nice video|0|0|1|2
3|Cringe|0|0|3|4
4|Interesting video|0|0|3|3
5|very good|0|0|7|2
6|another one|1|1|5|1
7|ahahahahahah|3|1|5|1
8|great match!!!!|4|0|7|1
9|Bro, you are crazy!|1|0|8|8
10|NULL|0|0|12|3
11|Poor Ethan|1|0|13|1
```

LogUser_Id	Id_Video	Local
4	5	Twitter
3	1	Tumblr
2	1	Facebook
1	1	Copy Link
1	2	Reddit
2	2	NULL
3	6	Facebook
4	6	LinkedIn
2	5	Facebook
1	7	Facebook
1	8	Facebook
4	13	NULL

```
sqlite> SELECT * from PARTILHA;
4|5|Twitter
3|1|Tumblr
2|1|Facebook
1|1|Copy Link
1|2|Reddit
2|2|NULL
3|6|Facebook
4|6|LinkedIn
2|5|Facebook
1|7|Facebook
1|8|Facebook
4|13|NULL
```