

Projeto de
Base de Dados
MIEIC - 2º Ano

YouTube

3ª Entrega

Ana Barros - up201806593
Davide Castro - up201806512
Henrique Ribeiro - up201806529

Turma 4 - Grupo 8

1 - Descrição do projeto

Com este projeto pretendemos desenvolver uma base de dados semelhante à usada para a plataforma YouTube™. O YouTube™ é uma plataforma de *streaming* de vídeos e, ao mesmo tempo, uma rede social. Embora seja necessário recorrer à simplificação do esquema devido à enorme complexidade desta plataforma, pretendemos elaborar os aspetos principais.

Para representar um utilizador criamos uma classe **User** identificado pelo seu ID e e-mail (único para todos os utilizadores), password, nome e se possui ou não o histórico de vídeos ativo.

Para além dos utilizadores normais, também incluímos os utilizadores *premium*, representados como **PremiumUser**, classe derivada da User, que possui uma mensalidade a ser paga para que estes possam ver vídeos sem anúncios associados.

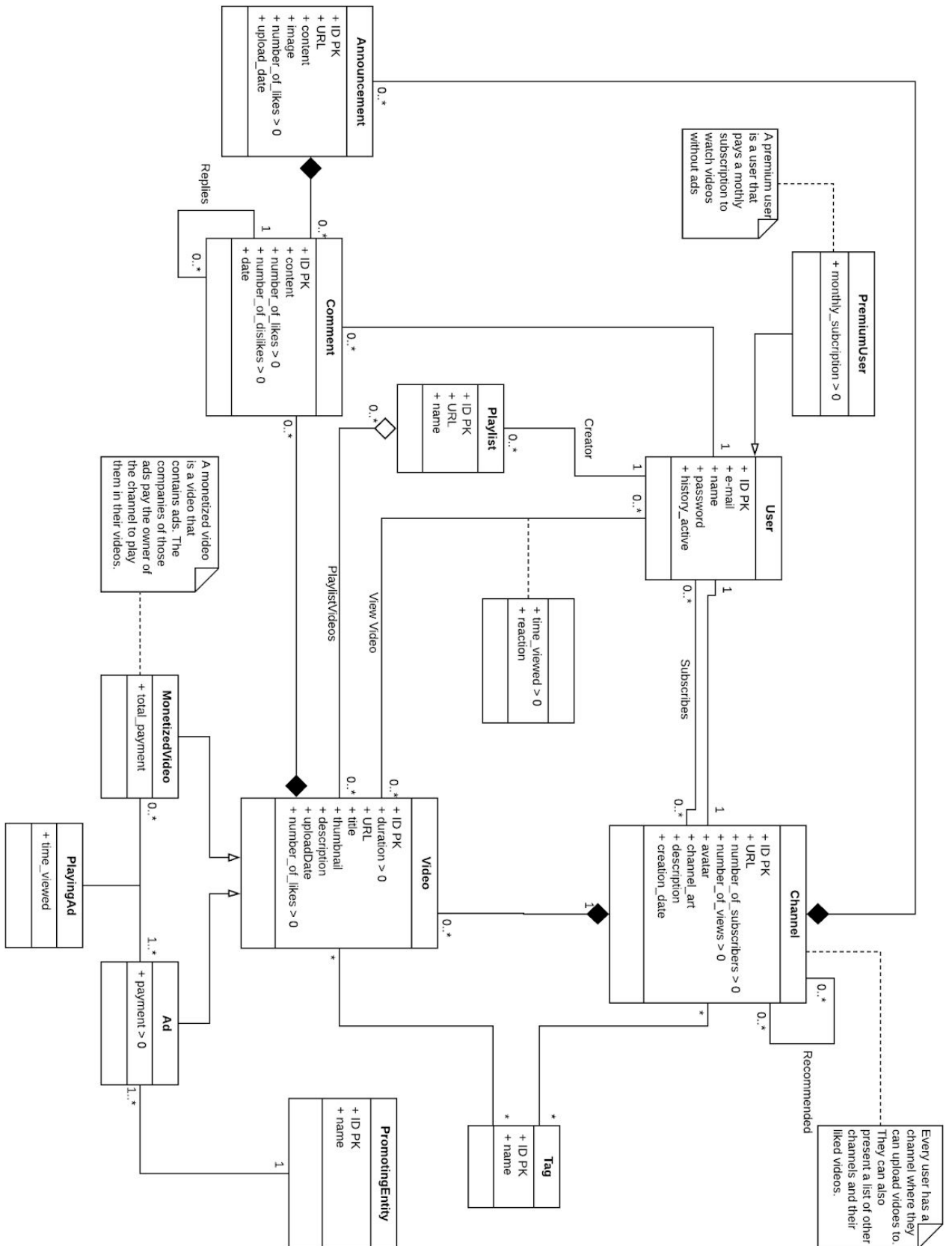
Cada User tem um canal representado por **Channel** responsável pela publicação de vídeos. Os canais têm um único URL, número de subscritores e visualizações (soma total do número de visualizações dos vídeos publicados pelo canal) e imagens para personalizar a página *web* do canal (avatar e channel_art). É a partir destes canais que são publicados vídeos.

Por sua vez, os vídeos, classe **Video**, são caracterizados pelo seu único URL, título, duração e *thumbnail* (imagem que é apresentada no ícone do vídeo). Estes vídeos podem ser monetizados (**MonetizedVideo**), ou seja, são passados anúncios, da classe **Ad**, também derivada de **Video**, que geram dinheiro para o canal. As companhias patrocinadoras dos vídeos são *promoting_entity* e pagam ao proprietário do canal para que publicidade seja transmitida nos vídeos. Os vídeos podem ser compilados por um utilizador para criar uma **Playlist** (lista de reprodução de vídeos) e, para cada vídeo, existe uma secção reservada a comentários (**Comment**). Para cada comentário, os utilizadores podem dar um “gosto”, “desgosto” ou criar o seu próprio tendo a hipótese de responder.

Tanto os **Channels** como os **Videos** têm uma relação em comum, a **Description**. Nesta classe é guardado um texto que sumarie o canal ou vídeo associado e pode conter **Tags** (palavras chave) que fazem com que o vídeo ou canal seja recomendado quando são procuradas essas *tags*.

Os canais podem não só publicar vídeos como também anúncios, representados pela classe **Announcement**, que contêm uma imagem, número de gostos dados pelos outros utilizadores, tal como os seus comentários, e também um texto.

2 - Diagrama UML



3 - Modelo Relacional

- **User**(ID, e-mail, name, password, history_active, monthly_subscription)
- **Channel**(ID, URL, number_of_subscribers, number_of_views, avatar, channel_art, description, creation_date, →User)
- **TagChannel**(idChannel→Channel, idTag→Tag)
- **Subscribes**(idUser→User, idChannel→Channel)
- **Recommended**(ID1→Channel, ID2→Channel)
- **Comment**(ID, content, number_of_likes, number_of_dislikes, date, IDannouncement→Announcement, IDvideo→Video, IDuser→User)
- **Replies**(IDReply→Comment, IDmainComment→Comment)
- **Announcement**(ID, URL, content, image, number_of_likes, upload_date, IDchannel→Channel)
- **Video**(ID, URL, duration, title, thumbnail, description, uploadDate, IDchannel→Channel)
- **TagVideo**(IDvideo→Channel, IDtag→Tag)
- **Playlist**(ID, URL, name, IDuser→User)
- **MonetizedVideo**(ID→Video, total_payment)
- **Ad**(ID→Video, payment, IDPromotingEntity→PromotingEntity)
- **PromotingEntity**(ID, name)
- **Tag**(ID, name)
- **ViewVideo**(time_viewed, reaction, IDvideo→Video, IDuser→User)
- **PlaylistVideos**(IDplaylist→Playlist, IDvideo→Video)

4 - Dependências Funcionais e Formas Normais

User: ID→e-mail, name, password, history_active, monthly_subscription
e-mail→ID

Chaves da relação: {ID, e-mail};

. Esta relação está na forma BCNF porque para cada dependência funcional não trivial, o lado esquerdo é uma super chave.

Channel: ID→URL, number_of_subscribers, number_of_views, avatar, channel_art, description, userID, creation_date
userID→URL

Chaves da relação: {ID, userID};

. Esta relação está na forma BCNF porque para cada dependência funcional não trivial, o lado esquerdo é uma super chave.

Comment: ID→content, IDannouncement, IDvideo, number_of_likes, number_of_dislikes, idUser, date

Chaves da relação: {ID};

. Esta relação está na forma BCNF porque para cada dependência funcional não trivial, o lado esquerdo é uma super chave.

Replies: idReply→idMainComment

Chaves da relação: {idReply};

. Esta relação está na forma BCNF porque a dependência funcional existente, não trivial, o lado esquerdo (idReply) é uma super chave.

Announcement:

ID→URL, content, image, number_of_likes, URLchannel

Chaves da relação: {ID};

. Esta relação está na forma BCNF porque para cada dependência funcional não trivial, o lado esquerdo é uma super chave

Video: ID→URL, IDchannel, tagName, title, duration, thumbnail, description

Chaves da relação: {ID};

. Esta relação está na forma BCNF porque a dependência funcional não trivial, o lado esquerdo é uma super chave.

Playlist: ID→URL, name, IDuser

Chaves da relação: {ID};

. Esta relação está na forma BCNF porque a única dependência funcional, que é não trivial, o lado esquerdo é uma super chave.

MonetizedVideo:

ID→total_payment

Chaves da relação: {ID};

. Esta relação está na forma BCNF porque a dependência funcional não trivial, o lado esquerdo é uma super chave.

Ad: ID→payment, name, IDPromotingEntity

Chaves da relação: {ID};

. Esta relação está na forma BCNF porque para cada dependência funcional não trivial, o lado esquerdo é uma super chave.

ViewVideo: IDvideo, IDuser→time_viewed, reaction

Chaves da relação: {{IDvideo, IDuser}};

. Esta relação está na forma BCNF porque a dependência funcional não trivial, o lado esquerdo é uma super chave.

É assim possível observar que em todas as dependências funcionais não-triviais de cada relação, o lado esquerdo é uma super-chave da relação, logo, estão em **cumprimento** com a Boyce-Codd Normal Form (**BCNF**) e, conseqüentemente, cumprem a 3ª Forma Normal (**3NF**).

6 - Restrições

6.1 - Restrições CHECK

1. User:

- password → A password tem que ter um comprimento maior que 5 caracteres por razões de segurança;
- history_active → Só pode ser 1 (histórico ativo) ou 0 (inativo);
- monthly_subscription → Não pode ter valor negativo e tem que ser menor que 200;

2. Channel:

- number_of_likes → Não pode ser negativo;
- number_of_dislikes → Não pode ser negativo;

3. Comment:

- content → O seu tamanho tem que ser maior que 0, pois é o corpo do comentário.
- number_of_likes → Não pode ser negativo;
- number_of_dislikes → Não pode ser negativo;

4. Video:

- duration → Tem que ser maior que 0 pois não pode existir um vídeo com duração nula.
- title → O seu tamanho tem que ser maior que 0 pois é aplicada uma obrigação de todos os vídeos conterem um título

5. MonetizedVideo:

- total_payment → O pagamento tem que ser maior que 0;

6. NumberOfTimesViewed:

- number_of_times_viewed → O total de visualizações não pode ser negativo;

7. ViewVideo:

- time_viewed → O tempo de visualização não pode ser negativo;
- reaction → Pode ser 0 (sem reação), 1 (gostou), -1 (não gostou);

8. Announcement:

- O comunicado tem que incluir um texto com tamanho maior que 0, uma imagem ou então ambos.

9. Recommended:

- O URL recomendado não pode ser o mesmo que o está a recomendar

6.2 - Restrições Chave (PRIMARY KEY ou UNIQUE)

1. User

- Não podem existir dois utilizadores com o mesmo ID;
- Não podem existir dois utilizadores com o mesmo e-mail;

2. Channel

- Não podem existir canais com o mesmo URL;

3. Comment

- Não podem existir comentários com o mesmo ID;

4. Announcement

- Não podem existir comunicados com o mesmo URL;

5. Video

- Não podem existir vídeos com o mesmo URL;

6. Playlist

- Não podem existir listas de reprodução com o mesmo URL;

7. PromotingEntity

- Não podem existir entidade promotora com o mesmo nome;

8. Tag

- Não podem existir *tags* com o mesmo nome;

6.3 - Restrições NOT NULL

1. User

- No utilizador nenhum dos campos (e-mail, name, password, history_active e monthly_subscription) pode ser nulo. O e-mail, o nome e a password são essenciais ao *login* do utilizador na plataforma. Por sua vez, o history_active tem que estar definido para se saber o comportamento a atuar na plataforma quanto ao histórico de visualizações e o monthly_subscription para o pagamento a efetuar pelo utilizador pelo pacote desejado.

2. Creation

- Nesta relação o campo createDate é obrigatoriamente não-nulo, pois a data de criação é um elemento necessário para o registo da criação de um canal.

3. Channel

- Em cada canal é obrigatório os campos number_of_subscribers, number_of_views e avatar não serem definidos como NULL, pois são elementos que devem estar sempre definidos num canal, tendo estes um valor default a 0 para os números e uma imagem predefinida para o avatar.

4. Comment

- A data nunca pode ser NULL em caso nenhum, pois tem que ser uma data válida.

5. Announcement

- O valor number_of_likes não pode ser nulo, uma vez que existe em todos os comunicados tendo o valor por default 0.

6. Commented

- Nesta relação, que define o ato de um utilizador comentar um video/announcement, o campo date não pode ser nulo pois a data é um valor essencial para o registo do comentário e a relação não teria qualquer uso sem esta.

7. Upload

- A data nunca pode ser NULL em caso nenhum, pois tem que ser uma data válida.

8. Video

- O título e a thumbnail de um vídeo tem que ser NOT NULL uma vez que este tem que ser identificado.
 - A duração de um vídeo também não pode ser NULL porque tem que ter uma duração existente e válida.

9. Playlist

- Na nossa implementação desta plataforma cada *playlist* precisa de ter, obrigatoriamente, um nome, pelo que o campo name nunca pode ser nulo.

10. MonetizedVideo

- O pagamento total de um MonetizedVideo não pode ser NULL porque se assim fosse seria um video normal.

11. Ad

- Como o objetivo dos Ad's é providenciar um pagamento aos canais que os inserem nos seus vídeos, o campo payment nunca poderá ser nulo.

12. NumberOfTimesViewed

- O número total de visualizações não pode ser NULL uma vez que existe sempre em qualquer vídeo sendo por default o valor 0.

13. ViewVideo

- O campo reaction não pode ser nulo, visto que se o utilizador não reagir ao vídeo nós definimos o valor de reaction a 0 como default. Também o time_viewed tem de ser preenchido obrigatoriamente para, por exemplo, permitir aos utilizadores serem direcionados para a parte do vídeo onde estavam na última sessão.

6.4 - Restrições de Integridade Referencial (chave estrangeira)

1. Subscribes

- ID é uma referência ao ID de um utilizador que está a subscrever ao canal;
- URL é uma referência ao URL de um canal que está a ser subscrito;

2. Creation

- O ID nesta relação é uma referência ao ID do utilizador que criou o canal;
- Por sua vez, o URL é chave primária e também refere-se ao URL do canal criado;

3. Channel

- O userID é uma referência ao ID do utilizador a quem pertence o canal;

4. Recommended

- O recommended contém dois URLs, um é o canal recomendado e o outro o canal que está a recomendar, e desta forma os URLs são referências ao URL do canal respectivo;

5. Comment

- O URLannouncement é uma referência ao URL de um Announcement que foi comentado;

6. Replies

- Nesta relação usamos idReply como chave primária e referência para o comentário que serve de resposta ao comentário inicial, cujo ID é referenciado por idMainComment.

7. Announcement

- No anúncio o URLchannel é referência para o URL do canal usado para a publicação.

8. Commented

- Esta relação guarda a informação que liga um utilizador ao seu comentário, desta forma é necessário um id de utilizador que referencia o ID do utilizador e um id de comentário que referencia o ID do comentário escrito.

9. Upload

- No ato do carregamento de um vídeo usamos o URLchannel e o URLvideo como referência para os URLs do canal para onde é carregado o vídeo e o URL desse mesmo vídeo.

10. Video

- Para o vídeo necessitamos de uma chave estrangeira para referência ao canal que publicou o video (URLchannel) e também o tagName para identificar a tag usada.

11. Playlist

- Numa playlist é guardado um ID que é uma referência para o ID do User que a criou;

12. MonetizedVideo

- Vídeo monetizado é uma espécie especial de vídeo em que o user que o publicou pode receber dinheiro com este. Deste modo possui um URL que é uma referência para o URL de um vídeo.

13. Ad

- O anúncio é um vídeo que será passado no início dos outros, e desta forma o seu URL é uma referência ao URL de um vídeo.

14. NumberOfTimesViewed

- Nesta relação são necessárias duas chaves estrangeiras: URLad como referência ao URL do anúncio a ser visualizado e URLmonetizedVideo, que refere o vídeo a ser monetizado por mostrar o anúncio.

15. ViewVideo

- Na relação de visualização de um vídeo usamos URL para referência ao vídeo a ser visualizado e o ID referente ao utilizador que o visualiza.

16. PlaylistVideos

- Esta relação guarda contém os vídeos que pertencem a uma certa playlist, desta forma tem como atributo dois URLs que se referenciam ao URL de um vídeo e a um URL de uma playlist.

17. AnnouncementUpload

- Nesta relação de carregamento de um comunicado são apenas necessárias duas chaves estrangeiras (URLchannel e URLannouncement), de forma a fazer referência tanto ao canal que fez o *upload* como ao comunicado a ser publicado.

7 - Interrogações

1. Lista das playlists que contêm os vídeos com as tags mais vistas nos últimos 60 dias;
2. Histórico de visualizações de cada utilizador, incluindo o título e ID dos vídeo visualizados, a data em que viu, o tempo de visualização dos vídeos e a sua reação, por ordem decrescente de User ID e da data de visualização;
3. Calcula o tempo total de vídeo em cada canal cujo user tem pelo menos um 'a' no seu nome;
4. Mostra os comentários existentes e se for um comentário a um comentário, mostra o ID do comentário a que está responder.
5. Lista os top influencers da plataforma. Isto é, lista os canais com maior impacto na plataforma (maior popularidade), avaliado pela média do número de subscritores, número de likes e número de vídeos de cada canal. Considera valores nulos como 0 para que apareçam todos os canais. Ordenado pela média;
6. Calcula as visualizações de cada vídeo e soma esse número por canal para obtermos número total de visualizações por canal. De seguida, fazemos a diferença desse número para os dois canais com mais visualizações;
7. Lista o ID e nome de todas as Promoting Entities e o respetivo pagamento total, tendo em conta as vezes em que cada Ad da entity foi visto, considerando o tempo em que o ad passou no vídeo com o tempo que os utilizadores pararam de ver o vídeo;
8. Lista os vídeos recomendados a um certo utilizador, tendo em conta os utilizadores com as mesmas preferências. Neste caso, para um dado utilizador procura o utilizador que têm as mesmas preferências que este. Isto é, procura o utilizador que tiver o máximo de vídeos vistos em comum com o outro. Este utilizador que encontramos passa a ser a Match do utilizador inicial. Depois disso, temos que encontrar os vídeos que não viram em comum e ir buscar os canais que publicaram esses vídeos. No final, recomendamos os vídeos desses canais exceto os que o utilizador já viu porque é provável que

se o utilizador Match viu vídeos desses canais, então o utilizador inicial irá gostar de ver vídeos desses canais também;

9. Ordena os canais por rendimento total de cada canal. Para o rendimento temos em conta o número de views e o número de likes só para os vídeos monetizados;
10. Para um utilizador, simula a secção de subscrições do youtube mas adicionalmente retirando os vídeos já vistos pelo utilizador. Ou seja, lista dos vídeos postados pelos canais a que está subscrito os que ainda não viu , ordenados pelos mais recentes;

8 - Gatilhos

1. Assegura de que. quando se insere um User novo. é automaticamente inserido um Channel para esse User, inicializado com uma imagem de perfil aleatória, com o nome e o do User, com data de criação no momento em que o User é criado e descrição vazia.
2. Quando um User subscreve um Channel, se este não for o dono do Channel subscrito, incrementa o número de subscritores desse channel por um. Caso um user tentar subscrever ao seu próprio canal um erro ocorre.
3. Quando um utilizador vê um vídeo, o número de likes do vídeo é atualizado consoante a reação do utilizador (dislike -1, like 1, sem reação 0). Além disso, se o vídeo for monetizado, o rendimento total desse vídeo é atualizado consoante o pagamento de cada Ad que passou dentro do tempo que o utilizador visualizou o vídeo.