

# Projeto Banco de Dados - **StreamVideo**

Alunos:

Breno Santos

Guilherme Ribeiro

Luca Neves

Repositório: [https://github.com/guilhermerccc/projeto\\_bd](https://github.com/guilhermerccc/projeto_bd)





## **1 - Minimundo**

A “StreamVideo” é uma plataforma de vídeos, onde usuários podem assistir conteúdos do seu interesse.

Um **Usuário** na plataforma possui um CPF, E-mail, Nome, Senha e Endereço, com CEP e Complemento. Todo usuário deve ser **Espectador** e/ou **Streamer**. Não existem usuários que não fazem parte de alguma das duas categorias e podem existir usuários que são espectadores e streamers ao mesmo tempo, em contas diferentes.

Um **Espectador**, além dos atributos presentes em Usuário, pode cadastrar categorias do seu interesse (pode existir mais de uma categoria). Um espectador pode **seguir** outros espectadores, a fim de encontrar interesses semelhantes. Um espectador pode seguir vários espectadores e pode também ser seguido por vários.

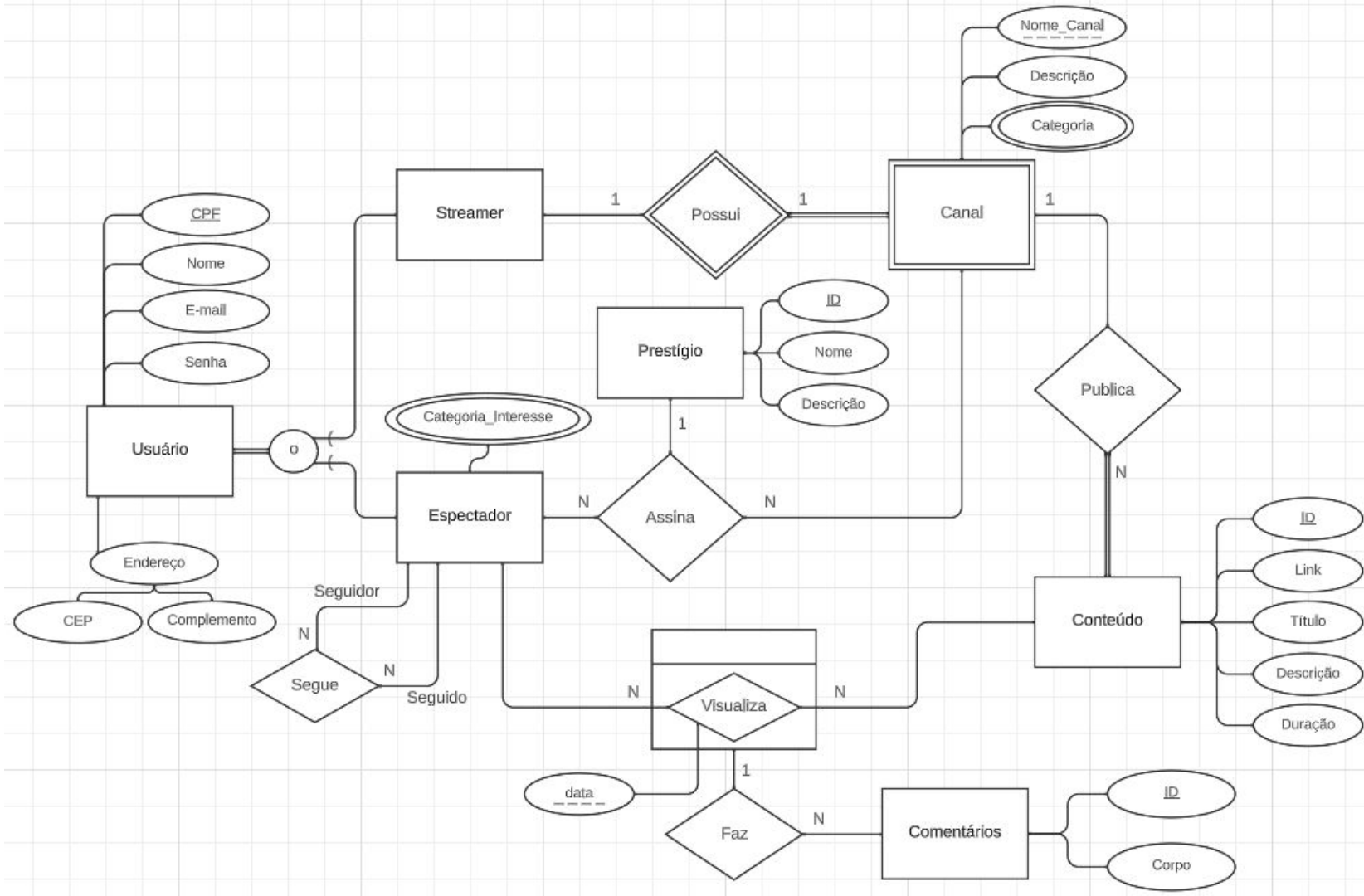
Um **Streamer** não tem nenhum atributo além dos já presentes no usuário. Um streamer, no entanto, possui um **Canal**, que será onde as relações com o Espectador acontecerão. O Canal só existe com a presença do Streamer e está associado ao seu CPF. Um Streamer possui um canal e um canal é possuído por apenas um streamer. Um Canal possui nome, descrição e categoria (pode ter mais de uma categoria).

Existe um relacionamento envolvendo o Espectador, o Canal e a **Prestígio** (que ele terá nesse canal ao realizar uma **assinatura**). Os espectadores de um canal devem assinar no máximo um prestígio (ID, nome, descrição), o qual pode ser assumido por mais de um espectador no mesmo canal. Um espectador pode assumir o mesmo prestígio em vários canais.

Além disso, um Canal pode publicar **Conteúdos**, que não existem sem os Canais. Um Canal pode publicar vários conteúdos, e cada conteúdo só pode ser publicado por um canal. O conteúdo possui um identificador, link, título, descrição e duração.

Por último, o Espectador pode **visualizar** o conteúdo. Um espectador pode visualizar vários conteúdos e um conteúdo pode ser visualizado por vários espectadores. A visualização possui uma data. Nesse processo de visualização, o espectador pode fazer **comentários** (ID), que possuem um corpo. Os comentários não são obrigatórios, mas só podem ser feitos se ocorrer uma visualização. Em uma visualização, podem ser feitos vários comentários, e cada comentário é feito em no máximo uma visualização.

## **2 - Modelo Conceitual**



3

## **3 - Modelo Lógico**

1. Espectador(CPF, Nome, Email, Senha, End\_Cep, End\_Complemento)
2. Categoria\_Interesse(CPF\_categoria\_interesse)  
CPF -> Espectador(CPF)
3. Segue(CPF\_Seguidor\_CPF\_Seguido)  
CPF\_Seguidor -> Espectador(CPF)  
CPF\_Seguido -> Espectador(CPF)
4. Streamer(CPF, Nome, Email, Senha, End\_Cep, End\_Complemento)
5. Canal(CPF\_Nome\_Canal, descrição)  
CPF -> Streamer(CPF)
6. Categoria(CPF\_Nome\_Canal\_categoria)  
(CPF, Nome\_Canal) -> Canal(CPF, Nome\_Canal)
7. Conteúdo(ID, Link, Título, Descrição, Duração, (CPF\_Streamers, Nome\_Canal)!)  
(CPF\_Streamers, Nome\_Canal) -> Canal(CPF, Nome\_Canal)
8. Visualiza(CPF\_ID\_data)  
CPF -> Espectador(CPF)  
ID -> Conteúdo(ID)
9. Comentários(ID, Corpo, (CPF\_ID\_conteúdo, data)!)  
(CPF\_ID\_conteúdo, data) -> Visualiza(CPF, ID, data)
10. Prestigio (ID, Nome, Descrição)
11. Assina(CPF\_Esp\_CPF\_Str\_Nome\_Canal\_ID\_Prestigio)  
CPF\_Esp -> Espectador(CPF)  
(CPF\_Str, Nome\_Canal) -> Canal(CPF, Nome\_Canal)  
ID\_Prestigio -> Prestigio(ID)

## 4.1 - Criação.SQL



```
-- CRIA TABELA ESPECTADOR
CREATE TABLE ESPECTADOR(
    CPF NUMBER(11),
    NOME VARCHAR(80) NOT NULL,
    EMAIL VARCHAR(255) NOT NULL UNIQUE,
    SENHA VARCHAR(15) NOT NULL,
    END_CEP NUMBER(8),
    END_COMPLEMENTO VARCHAR(255),
    CONSTRAINT PK_ESPECTADOR PRIMARY KEY (CPF)
);

-- CRIA TABELA CATEGORIA INTERESSE
CREATE TABLE CATEGORIA_INTERESSE(
    CPF NUMBER(11),
    CATEGORIA_INTERESSE VARCHAR(255),
    CONSTRAINT PK_CATEGORIA_INTERESSE PRIMARY KEY (CPF, CATEGORIA_INTERESSE),
    CONSTRAINT FK_CATEGORIA_INTERESSE FOREIGN KEY (CPF) REFERENCES ESPECTADOR(CPF) ON DELETE CASCADE
);

-- CRIA TABELA SEGUE
CREATE TABLE SEGUE(
    CPF_SEGUIDOR NUMBER(11),
    CPF_SEGUIDO NUMBER(11),
    CONSTRAINT PK_SEGUE PRIMARY KEY (CPF_SEGUIDOR, CPF_SEGUIDO),
    CONSTRAINT FK_CPF_SEGUIDOR FOREIGN KEY (CPF_SEGUIDOR) REFERENCES ESPECTADOR(CPF) ON DELETE CASCADE,
    CONSTRAINT FK_CPF_SEGUIDO FOREIGN KEY (CPF_SEGUIDO) REFERENCES ESPECTADOR(CPF) ON DELETE CASCADE
);

-- CRIA TABELA STREAMER
CREATE TABLE STREAMER(
    CPF NUMBER(11),
    NOME VARCHAR(80) NOT NULL,
    EMAIL VARCHAR(255) NOT NULL UNIQUE,
    SENHA VARCHAR(15) NOT NULL,
    END_CEP NUMBER(8),
    END_COMPLEMENTO VARCHAR(255),
    CONSTRAINT PK_STREAMER PRIMARY KEY (CPF)
);
```

```

-- CRIA TABELA CANAL
CREATE TABLE CANAL(
    CPF NUMBER(11),
    NOME_CANAL VARCHAR(20) NOT NULL,
    DESCRICAO VARCHAR(255),
    CONSTRAINT PK_CANAL PRIMARY KEY (CPF, NOME_CANAL),
    CONSTRAINT FK_CANAL FOREIGN KEY (CPF) REFERENCES STREAMER(CPF) ON DELETE CASCADE
);

-- CRIA TABELA CATEGORIA
CREATE TABLE CATEGORIA(
    CPF NUMBER(11),
    NOME_CANAL VARCHAR(20),
    CATEGORIA VARCHAR(20),
    CONSTRAINT PK_CATEGORIA PRIMARY KEY (CPF, NOME_CANAL, CATEGORIA),
    CONSTRAINT FK_CATEGORIA FOREIGN KEY (CPF, NOME_CANAL) REFERENCES CANAL(CPF, NOME_CANAL) ON DELETE CASCADE
);

--CRIA TABELA CONTEÚDO
CREATE TABLE CONTEUDO(
    ID NUMBER(8),
    LINK VARCHAR(255) NOT NULL UNIQUE,
    TITULO VARCHAR(255) NOT NULL,
    DESCRICAO VARCHAR(255),
    DURACAO NUMBER NOT NULL,
    CPF_STREAMER NUMBER(11) NOT NULL,
    NOME_CANAL VARCHAR(20) NOT NULL,
    CONSTRAINT PK_CONTEUDO PRIMARY KEY (ID),
    CONSTRAINT FK_CONTEUDO FOREIGN KEY (CPF_STREAMER, NOME_CANAL) REFERENCES CANAL(CPF, NOME_CANAL) ON DELETE CASCADE
);

--CRIA TABELA VISUALIZA
CREATE TABLE VISUALIZA(
    CPF NUMBER(11),
    ID NUMBER(8),
    DATA_VISUALIZACAO TIMESTAMP(3) NOT NULL,
    CONSTRAINT PK_VISUALIZA PRIMARY KEY (CPF, ID, DATA_VISUALIZACAO),
    CONSTRAINT FK_VISUALIZA_CPF FOREIGN KEY (CPF) REFERENCES ESPECTADOR(CPF),
    CONSTRAINT FK_VISUALIZA_ID FOREIGN KEY (ID) REFERENCES CONTEUDO(ID) ON DELETE CASCADE
);

```

```
-- CRIA TABELA COMENTÁRIOS
CREATE TABLE COMENTARIOS(
    ID NUMBER(8),
    CORPO VARCHAR(255),
    CPF NUMBER(11) NOT NULL,
    ID_CONTEUDO NUMBER(8) NOT NULL,
    DATA_VISUALIZACAO TIMESTAMP NOT NULL,
    CONSTRAINT PK_COMENTARIOS PRIMARY KEY (ID),
    CONSTRAINT FK_COMENTARIOS FOREIGN KEY (CPF, ID_CONTEUDO, DATA_VISUALIZACAO) REFERENCES VISUALIZA(CPF, ID, DATA_VISUALIZACAO) ON DELETE CASCADE
);

--CRIA TABELA PRESTÍGIO
CREATE TABLE PRESTIGIO(
    ID NUMBER(4),
    NOME VARCHAR(255),
    DESCRICAO VARCHAR(255),
    CONSTRAINT PK_PRESTIGIO PRIMARY KEY (ID)
);

--CRIA TABELA ASSINA
CREATE TABLE ASSINA(
    CPF_ESP NUMBER(11),
    CPF_STR NUMBER(11),
    NOME_CANAL VARCHAR(20),
    ID_PRESTIGIO NUMBER(4) NOT NULL,
    CONSTRAINT PK_ASSINA PRIMARY KEY (CPF_ESP, CPF_STR, NOME_CANAL),
    CONSTRAINT FK_ASSINA_ESP FOREIGN KEY (CPF_ESP) REFERENCES ESPECTADOR(CPF) ON DELETE CASCADE,
    CONSTRAINT FK_ASSINA_STR FOREIGN KEY (CPF_STR, NOME_CANAL) REFERENCES CANAL (CPF, NOME_CANAL) ON DELETE CASCADE,
    CONSTRAINT FK_ASSINA_PRE FOREIGN KEY (ID_PRESTIGIO) REFERENCES PRESTIGIO(ID) ON DELETE CASCADE
);
```

## 4.2 - Povoamento.SQL

```
-- INSERE ESPECTADORES
INSERT INTO ESPECTADOR VALUES (11112222333, 'Pedro', 'pedro@email.com', 'senha123', 54321000, 'Apartamento 101');
INSERT INTO ESPECTADOR VALUES (44445555666, 'Maria', 'maria@email.com', 'senha456', 12345678, 'Casa 202');
INSERT INTO ESPECTADOR VALUES (66665555777, 'Lucas', 'lucas@email.com', 'senha789', 65432109, 'Casa 505');
INSERT INTO ESPECTADOR VALUES (33332222888, 'Marta', 'marta@email.com', 'senha012', 56789012, 'Apartamento 606');
INSERT INTO ESPECTADOR VALUES (11113333999, 'Paulo', 'paulo@email.com', 'senha234', 54321234, 'Apartamento 707');
INSERT INTO ESPECTADOR VALUES (55557777888, 'Luiza', 'luiza@email.com', 'senha567', 98762345, 'Apartamento 010');
INSERT INTO ESPECTADOR VALUES (88889999000, 'Gustavo', 'gustavo@email.com', 'senha890', 76543210, 'Casa 111');
INSERT INTO ESPECTADOR VALUES (44443333222, 'Larissa', 'larissa@email.com', 'senha123', 23456789, 'Casa 121');
INSERT INTO ESPECTADOR VALUES (99996666777, 'Felipe', 'felipe@email.com', 'senha456', 11223344, 'Apartamento 131');
INSERT INTO ESPECTADOR VALUES (11119999222, 'Thiago', 'thiago@email.com', 'senha789', 11223346, 'Apartamento 151');
INSERT INTO ESPECTADOR VALUES (11119999233, 'Guilherme', 'guilherme@email.com', 'senha012', 11223355, 'Apartamento 102');
INSERT INTO ESPECTADOR VALUES (17899922333, 'Breno', 'breninho@gmail.com', 'breno123', 50720645, 'Casa 11A');
```

```
-- INSERE CATEGORIA DE INTERESSE
INSERT INTO CATEGORIA_INTERESSE VALUES (11112222333, 'Tecnologia');
INSERT INTO CATEGORIA_INTERESSE VALUES (11112222333, 'Esportes');
INSERT INTO CATEGORIA_INTERESSE VALUES (11112222333, 'Moda');
INSERT INTO CATEGORIA_INTERESSE VALUES (44445555666, 'Esportes');
INSERT INTO CATEGORIA_INTERESSE VALUES (44445555666, 'Tecnologia');
INSERT INTO CATEGORIA_INTERESSE VALUES (55557777888, 'Moda');
INSERT INTO CATEGORIA_INTERESSE VALUES (44443333222, 'Viagens');
INSERT INTO CATEGORIA_INTERESSE VALUES (11119999222, 'Música');
INSERT INTO CATEGORIA_INTERESSE VALUES (17899922333, 'Música');
```

```
-- INSERE SEGUE
INSERT INTO SEGUE VALUES (11112222333, 44445555666);
INSERT INTO SEGUE VALUES (11112222333, 66665555777);
INSERT INTO SEGUE VALUES (44445555666, 11112222333);
INSERT INTO SEGUE VALUES (44445555666, 55557777888);
INSERT INTO SEGUE VALUES (66665555777, 11112222333);
INSERT INTO SEGUE VALUES (33332222888, 55557777888);
INSERT INTO SEGUE VALUES (55557777888, 44445555666);
INSERT INTO SEGUE VALUES (55557777888, 11112222333);
INSERT INTO SEGUE VALUES (44443333222, 11112222333);
INSERT INTO SEGUE VALUES (17899922333, 11119999233);
INSERT INTO SEGUE VALUES (11119999233, 17899922333);
```

```
-- INSERE STREAMERS
INSERT INTO STREAMER VALUES (77778888999, 'João', 'joao@email.com', 'senha123', 87654321, 'Apartamento 303');
INSERT INTO STREAMER VALUES (99998888777, 'Ana', 'ana@email.com', 'senha456', 98765432, 'Casa 404');
INSERT INTO STREAMER VALUES (77776666222, 'Carla', 'carla@email.com', 'senha789', 87651234, 'Casa 808');
INSERT INTO STREAMER VALUES (22223333444, 'Ricardo', 'ricardo@email.com', 'senha012', 12349876, 'Casa 909');
INSERT INTO STREAMER VALUES (22223333487, 'Pedro', 'pedro@email.com', 'senha345', 12349879, 'Casa 302');
INSERT INTO STREAMER VALUES (17899922333, 'Breno', 'breninho@gmail.com', 'breno123', 50720645, 'Casa 11A');
INSERT INTO STREAMER(CPF, NOME, EMAIL, SENHA) VALUES (11119999233, 'Guilherme', 'guilherme2@email.com', 'abcde');
INSERT INTO STREAMER(CPF, NOME, EMAIL, SENHA) VALUES (65239865412, 'José', 'jose@email.com', 'zezinho123');
```

Povoamento de  
algumas tabelas

## 4.3 - Consultas.SQL



-- 1. Liste o cpf e o nome dos espectadores que seguem mais de 1 pessoa (HAVING, GROUP BY, INNER JOIN)

```
SELECT S.CPF_SEGUIDOR, E.NOME, COUNT(*) as NUM_SEGUIDOS
FROM SEGUE S
INNER JOIN ESPECTADOR E
ON S.CPF_SEGUIDOR = E.CPF
GROUP BY S.CPF_SEGUIDOR, E.NOME
HAVING COUNT(*) > 1;
```

-- 2. Encontre os streamers que não possuem canais (SUBCONSULTA TIPO TABELA)

```
SELECT * FROM STREAMER
WHERE CPF NOT IN (SELECT CPF FROM CANAL);
```

-- 3. Qual é a média de duração dos conteúdos publicados por cada canal, juntamente com seu nome e sua descrição? (INNER JOIN)

```
SELECT C.NOME_CANAL, C.DESCRICAO, AVG(CT.DURACAO) AS MEDIA_DURACAO
FROM CANAL C
INNER JOIN CONTEUDO CT ON C.CPF = CT.CPF_STREAMER AND C.NOME_CANAL = CT.NOME_CANAL
GROUP BY C.NOME_CANAL, C.DESCRICAO;
```

-- 4. Projete o nome e o CPF dos espectadores que não assinam nenhum canal (LEFT OUTER JOIN)

```
SELECT E.CPF, E.NOME
FROM ESPECTADOR E LEFT OUTER JOIN ASSINA A ON E.CPF = A.CPF_ESP
WHERE A.CPF_STR IS NULL;
```

-- 5. Projetar o título e o número de visualizações do conteúdo com id = 2 (SUBCONSULTA TIPO ESCALAR)

```
SELECT C.TITULO, (SELECT COUNT(*) FROM VISUALIZA V WHERE V.ID = C.ID) AS NUM_VISUALIZACOES
FROM CONTEUDO C
WHERE C.ID = 2;
```

-- 6. Projetar o nome dos espectadores que assinam o Canal de João (SUBCONSULTA TIPO TABELA)

```
SELECT E.NOME AS NOME_ESPECTADOR
FROM ESPECTADOR E
WHERE E.CPF IN (
    SELECT A.CPF_ESP
    FROM ASSINA A
    WHERE A.NOME_CANAL = 'Canal de João'
);
```





```
SELECT E.NOME
FROM ESPECTADOR E
WHERE E.CPF IN (SELECT CI.CPF
                FROM CATEGORIA INTERESSE CI);
```

```
SELECT S.NOME
FROM STREAMER S
WHERE S.CPF IN (SELECT C.CPF FROM CANAL C)
AND S.END CEP IS NULL OR S.END COMPLEMENTO IS NULL;
```

```
SELECT E.CPF
FROM ESPECTADOR E
WHERE NOT EXISTS (
    SELECT *
    FROM SEQUE S
    WHERE S.CPF_SEGUIDOR = E.CPF
)
AND NOT EXISTS (
    SELECT *
    FROM ASSINA A
    WHERE A.CPF_ESP = E.CPF
);
```

```
SELECT E.NOME  
FROM ESPECTADOR E INNER JOIN ASSINA A ON E.CPF = A.CPF_ESP  
WHERE (A.CPF_STR, A.NOME_CANAL) = (SELECT C.CPF, C.NOME_CANAL  
FROM CANAL C INNER JOIN STREAMER S ON C.CPF = S.CPF  
WHERE S.EMAIL = 'breninho@gmail.com');
```

-- 15. Projete o nome dos espectadores que visualizaram um conteúdo ou assinaram um canal (UNION)

```
SELECT E.NOME FROM ESPECTADOR E
WHERE E.CPF IN (
    SELECT V.CPF FROM VISUALIZA V
    UNION
    SELECT A.CPF_ESP FROM ASSINA A
);
```

--16. Projete os cpfs dos usuários possuem contas de espectador e streamers, mas possuem o mesmo email nas duas contas

```
SELECT E.CPF, E.EMAIL
FROM ESPECTADOR E
INTERSECT
SELECT S.CPF, S.EMAIL
FROM STREAMER S
```

---

--17. Identifique quais espectadores comentaram mais de uma vez no mesmo conteúdo(GROUP BY, HAVING)

```
SELECT C.CPF, C.ID_CONTEUDO, COUNT(*) AS QUANTIDADE
FROM COMENTARIOS C
GROUP BY C.CPF, C.ID_CONTEUDO
HAVING COUNT(*) > 1;
```

--18. Identifique todos os canais que não possuem nenhum assinante (MINUS)

```
SELECT C.CPF, C.NOME_CANAL
FROM CANAL C
MINUS
SELECT A.CPF_STR, A.NOME_CANAL
FROM ASSINA A;
```



**Obrigado!**