

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
RIO GRANDE DO NORTE

Alunas: Maria Paula Felix do Santos Cunha
Jenyffer Danily de Souza Araújo

Documento de atividade de Banco de Dados

IFRN CAMPUS NATAL CENTRAL
2025

. O problema que se propõe a resolver:

- A dificuldade em acompanhar séries e quadros organizados no YouTube devido ao grande volume de conteúdo.

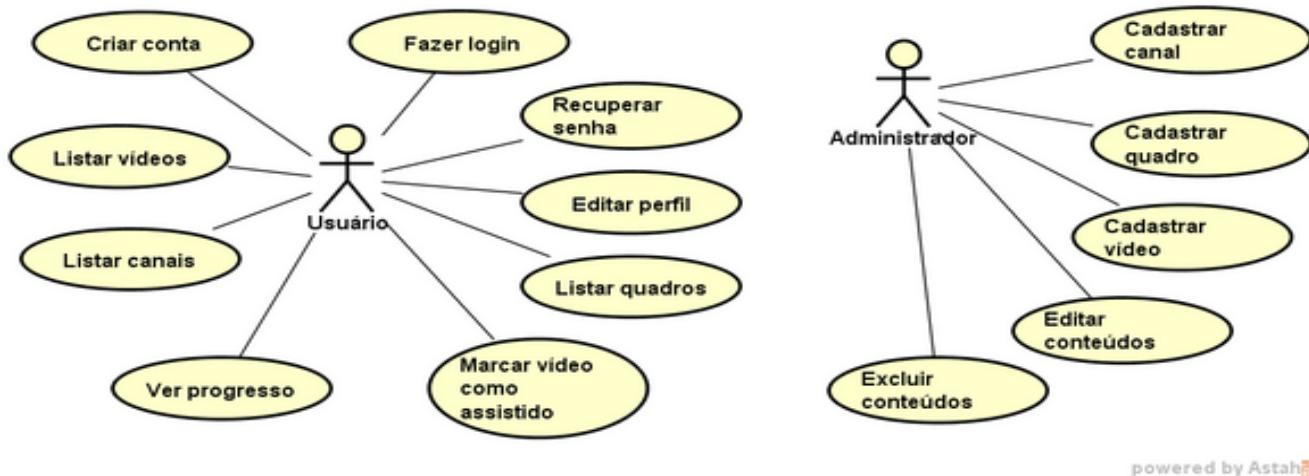
. Pessoas que sofrem com aquele problema:

- Pessoas que acompanham vídeos de quadros específicos,
- Fãs de canais educativos, podcasts, séries, etc,
- Criadores de conteúdo que desejam oferecer organização ao público.

. A solução Proposta:

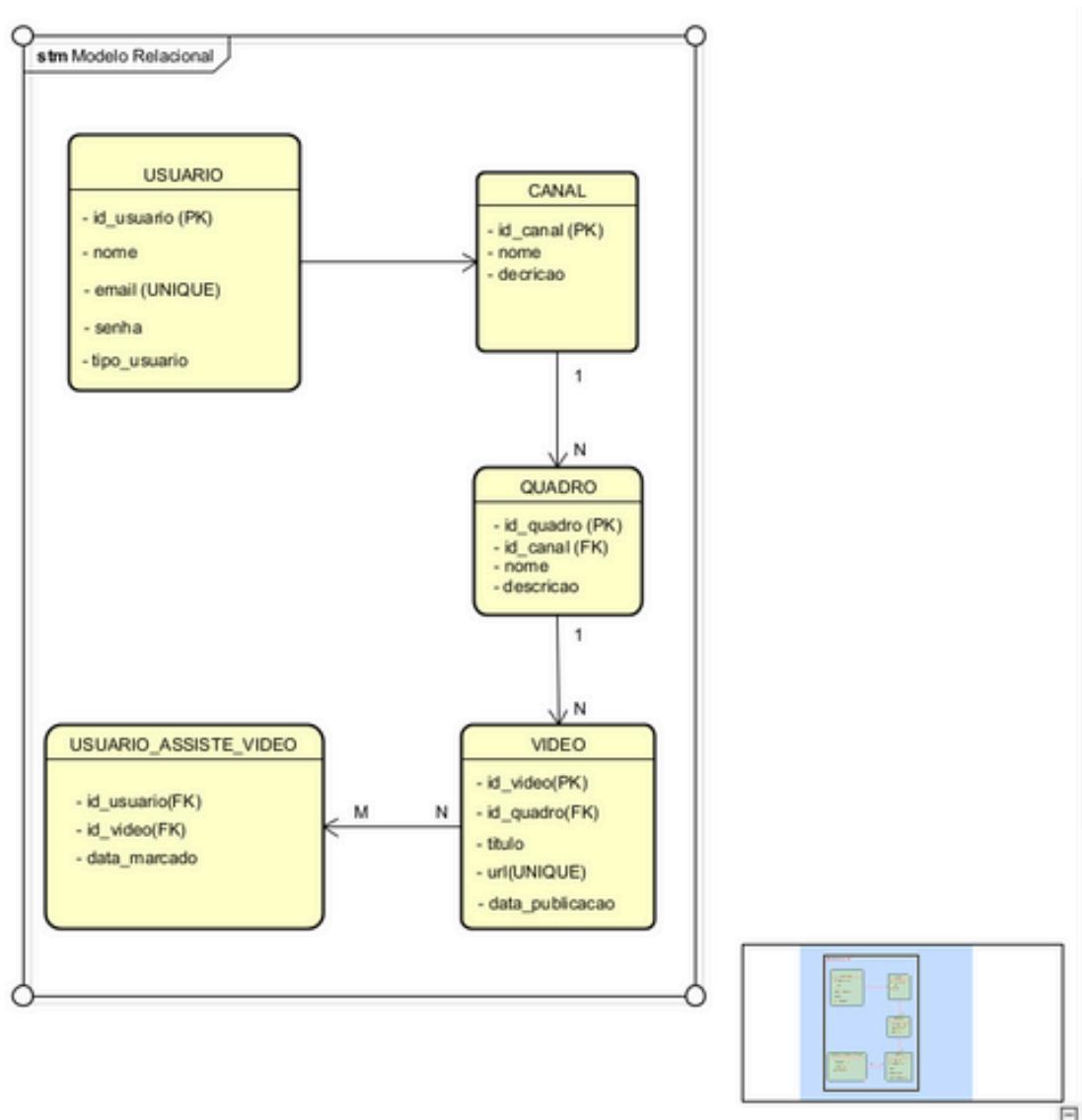
- O YTTrack visa organizar e facilitar o consumo de conteúdo do YouTube, permitindo:
 - Organizar vídeos por quadro e canal,
 - Marcar vídeos assistidos e acompanhar o progresso,
 - Administradores cadastrarem canais, quadros e vídeos (links do YouTube).

. Diagrama de Casos de Uso (CDU) com indicação dos CDUs implementados:



OBS: ainda não foi implementado nenhum diagrama porque o projeto é recente e está em desenvolvimento.

. Modelo Relacional (diagrama) sem as Pks compostas (exceto as tabelas intermediárias):



. Dicionário de Dados do Modelo Relacional do Item(e):

Tabela: USUARIO

Campo	Tipo	Tam	Nulo?	PK	UNIQUE	Descrição
Id_usuario	int	-	Não	s	-	Identificador do usuário
nome	varchar	100	Não	-	-	Nome do usuário
email	varchar	150	Não	-	s	email
senha	varchar	255	Não	-	-	Senha criptografada
tipo_usuario	VARCHAR	10	Não	-	-	Comum/admin

Tabela: CANAL

Campo	Tipo	Nulo?	PK
Id_canal	int	Não	s
nome	varchar	Não	-
descrição	text	Sim	-

Tabela: QUADRO

Campo	Tipo	Nulo?	PK	FK
Id_quadro	int	Não	s	-
Id_canal	int	Não	-	s
nome	Varchar(100)	Não	-	-
descrição	text	Não	-	-

Tabela: VIDEO

Campo	Tipo	Nulo?	PK	UNIQUE	FK
Id_video	int	Não	s	-	-
Id_quadro	int	Não	-	-	s
titulo	Varchar(200)	Não	-	-	-
url	Varchar(255)	Não	UNIQUE po quadro	-	
Data_publicacao	date	Sim	-	-	-

Tabela: USUARIO_ASSISTE_VIDEO

Campo	Tipo	Nulo?	PK	FK
Id_usuario	int	Não	S*	s
Is_video	int	Não	S*	s
Data_marcado	DATETIME	Não	-	-

OBS: s = sim nas tabelas

. Comandos CREATE TABLE das Tabelas do Projeto (Arquivo SQL):

```
CREATE TABLE usuario (
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(150) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    tipo_usuario ENUM('comum','admin') NOT NULL DEFAULT 'comum'
) ENGINE=InnoDB;
```

```
CREATE TABLE canal (
    id_canal INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    descricao TEXT
) ENGINE=InnoDB;
```

```
CREATE TABLE quadro (
    id_quadro INT AUTO_INCREMENT PRIMARY KEY,
    id_canal INT NOT NULL,
```

```
nome VARCHAR(100) NOT NULL,  
descricao TEXT,  
FOREIGN KEY (id_canal) REFERENCES canal(id_canal)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE video (  
    id_video INT AUTO_INCREMENT PRIMARY KEY,  
    id_quadro INT NOT NULL,  
    titulo VARCHAR(200) NOT NULL,  
    url VARCHAR(255) NOT NULL,  
    data_publicacao DATE,  
    UNIQUE (id_quadro, url),  
    FOREIGN KEY (id_quadro) REFERENCES quadro(id_quadro)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

```
CREATE TABLE usuario_assiste_video (  
    id_usuario INT NOT NULL,  
    id_video INT NOT NULL,  
    data_marcado DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (id_usuario, id_video),  
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    FOREIGN KEY (id_video) REFERENCES video(id_video)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
) ENGINE=InnoDB;
```

INSERTS:

-- Insere 5 registros na tabela usuario

```
INSERT INTO usuario (nome, email, senha, tipo_usuario) VALUES
('Maria Paula', 'maria@email.com', 'senha123', 'comum'),
('Jenyffer', 'jenyffer@email.com', 'senha456', 'comum'),
('Administrador', 'admin@email.com', 'admin123', 'admin'),
('Usuario1', 'usuario1@email.com', 'senha789', 'comum'),
('Usuario2', 'usuario2@email.com', 'senha101', 'comum');
```

-- Insere 5 registros na tabela canal

```
INSERT INTO canal (nome, descricao) VALUES
('Canal de Educação', 'Vídeos educativos sobre diversos temas'),
('Canal de Música', 'Músicas e clipes musicais'),
('Canal de Podcast', 'Podcasts sobre diversos assuntos'),
('Canal de Séries', 'Séries e filmes'),
('Canal de Games', 'Vídeos de jogos');
```

-- Insere 5 registros na tabela quadro

```
INSERT INTO quadro (id_canal, nome, descricao) VALUES
(1, 'Matemática', 'Vídeos sobre matemática básica'),
(1, 'História', 'Vídeos sobre história do Brasil'),
(2, 'Pop', 'Músicas pop atuais'),
(3, 'Entrevistas', 'Entrevistas com pessoas interessantes'),
(4, 'Comédia', 'Episódios de comédia');
```

-- Insere 5 registros na tabela video

```
INSERT INTO video (id_quadro, titulo, url, data_publicacao) VALUES
(1, 'Matemática Básica', 'url1', '2024-01-01'),
```

```
(2, 'História do Brasil', 'url2', '2024-01-02'),  
(3, 'Música Pop', 'url3', '2024-01-03'),  
(4, 'Entrevista', 'url4', '2024-01-04'),  
(5, 'Comédia', 'url5', '2024-01-05');
```

-- Insere 5 registros na tabela usuario_assiste_video

```
INSERT INTO usuario_assiste_video (id_usuario, id_video, data_marcado) VALUES  
(1, 1, '2024-01-06'),  
(1, 2, '2024-01-07'),  
(2, 3, '2024-01-08'),  
(3, 4, '2024-01-09'),  
(4, 5, '2024-01-10');
```

DML:

-- (a) Atualiza o tipo de usuário para 'admin' em todos os registros da tabela usuario

```
UPDATE usuario SET tipo_usuario = 'admin';
```

-- (b) Atualiza a descrição do canal para 'Canal de vídeos variados' onde o nome do canal é 'Canal de Educação'

```
UPDATE canal SET descricao = 'Canal de vídeos variados' WHERE nome = 'Canal de Educação';
```

-- (c) Atualiza o nome e a descrição do quadro onde o id_quadro é 1 e o id_canal é 1

```
UPDATE quadro SET nome = 'Matemática Avançada', descricao = 'Vídeos sobre matemática avançada' WHERE  
id_quadro = 1 AND id_canal = 1;
```

-- (d) Atualiza o título e a url do vídeo onde o id_video é 1

```
UPDATE video SET titulo = 'Matemática para Iniciantes', url = 'nova_url' WHERE id_video = 1;
```

-- (e) Adiciona '9' ao início do nome de todos os usuários (simulando um telefone)

```
UPDATE usuario SET nome = '9' + nome;
```

-- (f) Atualiza a data de publicação do vídeo para a data atual usando a função CURRENT_DATE

```
UPDATE video SET data_publicacao = CURRENT_DATE WHERE id_video = 1;
```

-- (g) Remove todos os registros da tabela usuario_assiste_video

```
DELETE FROM usuario_assiste_video;
```

-- (h) Remove o canal onde o nome é 'Canal de Música'

```
DELETE FROM canal WHERE nome = 'Canal de Música';
```

-- (i) Remove o quadro onde o id_quadro é 2 e o id_canal é 1

```
DELETE FROM quadro WHERE id_quadro = 2 AND id_canal = 1;
```

-- (j) Remove usuários com base em uma função agregada (ex: remover usuários com id_usuario menor que a média)

```
DELETE FROM usuario WHERE id_usuario < (SELECT AVG(id_usuario) FROM usuario);
```

SELECT:

-- (a) Lista todos os campos de todos os registros da tabela usuario

```
SELECT * FROM usuario;
```

-- (b) Lista o nome e o email dos usuários onde o tipo_usuario é 'comum'

```
SELECT nome, email FROM usuario WHERE tipo_usuario = 'comum';
```

-- (c) Lista o título e a url dos vídeos onde o id_quadro é 1 e a data_publicacao é anterior a '2024-01-05'

```
SELECT titulo, url FROM video WHERE id_quadro = 1 AND data_publicacao < '2024-01-05';
```

-- (d) Lista a quantidade de vídeos por quadro

```
SELECT id_quadro, COUNT(*) FROM video GROUP BY id_quadro;
```

-- (e) Lista os usuários que assistiram a vídeos do quadro com id_quadro = 1 (SELECTs aninhados)

```
SELECT nome FROM usuario WHERE id_usuario IN (SELECT id_usuario FROM usuario_assiste_video WHERE id_video IN (SELECT id_video FROM video WHERE id_quadro = 1));
```

junções:

-- (a) Três comandos SELECT com junção interna (INNER JOIN)

-- Lista o nome do usuário e o título do vídeo que ele assistiu

```
SELECT u.nome, v.titulo FROM usuario u INNER JOIN usuario_assiste_video uav ON u.id_usuario = uav.id_usuario INNER JOIN video v ON uav.id_video = v.id_video;
```

-- Lista o nome do canal e o nome do quadro

```
SELECT c.nome, q.nome FROM canal c INNER JOIN quadro q ON c.id_canal = q.id_canal;
```

-- Lista o título do vídeo e o nome do quadro a que ele pertence

```
SELECT v.titulo, q.nome FROM video v INNER JOIN quadro q ON v.id_quadro = q.id_quadro;
```

-- (b) Dois comandos SELECT com junção externa parcial (LEFT ou RIGHT OUTER JOIN)

-- Lista todos os canais e seus respectivos quadros (se existirem)

```
SELECT c.nome, q.nome FROM canal c LEFT JOIN quadro q ON c.id_canal = q.id_canal;
```

-- Lista todos os quadros e seus respectivos vídeos (se existirem)

```
SELECT q.nome, v.titulo FROM quadro q RIGHT JOIN video v ON q.id_quadro = v.id_quadro;
```

-- (c) Um comando SELECT com junção externa total (FULL OUTER JOIN)

-- Lista todos os canais e quadros, mostrando todos os registros de ambas as tabelas

```
SELECT c.nome, q.nome FROM canal c FULL OUTER JOIN quadro q ON c.id_canal = q.id_canal;
```