

**CURSO BACHAREL EM**  **SISTEMAS DE INFORMAÇÃO**

**IGOR ALMENARA  
FILIPE SCANDIANI**

**Administração de Bancos de Dados: Entrega do projeto**

**Campos dos Goytacazes/RJ**

**2025**

# **Minimundo**

A Plataforma de Conteúdo por Assinatura FastDev é um sistema web que permite aos usuários que acessem vídeos exclusivos mediante pagamento.

O acesso pode ser adquirido através de assinatura mensal/anual ou por meio de pagamento único via Pix. Além disso, a plataforma oferecerá um modelo freemium, permitindo o acesso gratuito a determinados vídeos.

Para atrair novos clientes, a plataforma disponibilizará um teste grátis onde o usuário poderá assistir um vídeo específico por até 5 minutos.

Após esse tempo, ele será automaticamente redirecionado para a página de planos para que possa assinar e continuar assistindo o conteúdo completo. O usuário poderá favoritar vídeos para assisti-los posteriormente.

A monetização será gerenciada por um gateway de pagamento, permitindo transações seguras. O usuário poderá cancelar sua assinatura a qualquer momento, perdendo imediatamente o acesso à plataforma.

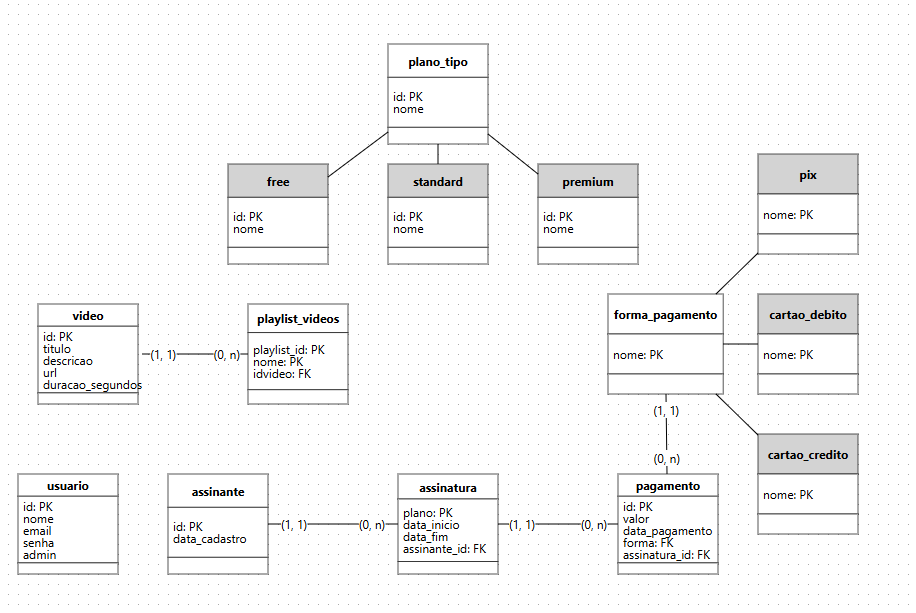
## 

## 

## 

## 

Modelo Lógico



Scripts DDL

CREATE TABLE usuario (

id SERIAL PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

senha VARCHAR(255) NOT NULL,

admin BOOLEAN NOT NULL

);

CREATE TABLE assinante (

id INT PRIMARY KEY REFERENCES usuario(id),

data\_cadastro DATE NOT NULL

);

CREATE TABLE plano\_tipo (

nome VARCHAR(20) PRIMARY KEY

);

INSERT INTO plano\_tipo (nome) VALUES

('STANDART'),

('PREMIUM'),

('FREE');

CREATE TABLE assinatura (

id SERIAL PRIMARY KEY,

plano VARCHAR(20) REFERENCES plano\_tipo(nome),

data\_inicio DATE NOT NULL,

data\_fim DATE NOT NULL,

ativa BOOLEAN NOT NULL,

assinante\_id INT REFERENCES assinante(id)

);

CREATE TABLE forma\_pagamento (

nome VARCHAR(30) PRIMARY KEY

);

INSERT INTO forma\_pagamento (nome) VALUES

('PIX'),

('CARTAO\_CREDITO'),

('CARTAO\_DEBITO');

CREATE TABLE pagamento (

id SERIAL PRIMARY KEY,

valor NUMERIC(10,2) NOT NULL,

data\_pagamento TIMESTAMP NOT NULL,

forma VARCHAR(30) REFERENCES forma\_pagamento(nome),

assinatura\_id INT UNIQUE REFERENCES assinatura(id)

);

CREATE TABLE playlist (

id SERIAL PRIMARY KEY,

titulo VARCHAR(100) NOT NULL

);

CREATE TABLE video (

id SERIAL PRIMARY KEY,

titulo VARCHAR(150) NOT NULL,

descricao TEXT,

url TEXT,

gratuito BOOLEAN DEFAULT FALSE,

duracao\_segundos INT

);

CREATE TABLE playlist\_videos (

playlist\_id INT REFERENCES playlist(id),

video\_id INT REFERENCES video(id),

PRIMARY KEY (playlist\_id, video\_id)

);

CREATE INDEX idx\_email\_usuario ON usuario(email);

CREATE INDEX idx\_gratuito\_video ON video(gratuito);

CREATE INDEX idx\_data\_inicio\_assinatura ON assinatura(data\_inicio);

Script DML

INSERT INTO usuario (nome, email, senha, admin) VALUES

('Filipe Scandiani', 'filipe@email.com', 'senha123', false),

('Maria Admin', 'maria@email.com', 'admin123', true);

INSERT INTO assinante (id, data\_cadastro) VALUES

(1, CURRENT\_DATE);

INSERT INTO assinatura (plano, data\_inicio, data\_fim, ativa, assinante\_id) VALUES

('MENSAL', CURRENT\_DATE, CURRENT\_DATE + INTERVAL '30 days', true, 1);

INSERT INTO pagamento (valor, data\_pagamento, forma, assinatura\_id) VALUES

(49.90, CURRENT\_TIMESTAMP, 'PIX', 1);

INSERT INTO video (titulo, descricao, url, gratuito, duracao\_segundos) VALUES

('Introdução ao Java', 'Aula gratuita introdutória', 'https://youtu.be/java123', true, 600),

('Spring Boot Avançado', 'Curso completo de Spring', 'https://fastdev.com/v/springboot', false, 1800);

INSERT INTO playlist (titulo) VALUES

('Playlist de Backend');

INSERT INTO playlist\_videos (playlist\_id, video\_id) VALUES

(1, 1),

(1, 2);

Script DCL

CREATE USER fastdev\_user WITH PASSWORD 'secure123';

GRANT CONNECT ON DATABASE fastdev TO fastdev\_user;

GRANT USAGE ON SCHEMA public TO fastdev\_user;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO fastdev\_user;

GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO fastdev\_user;

Previsão de relatórios

### Total de receita mensal agrupada por tipo de plano

SELECT

DATE\_TRUNC('month', p.data\_pagamento) AS mes,

a.plano,

SUM(p.valor) AS receita

FROM pagamento p

JOIN assinatura a ON p.assinatura\_id = a.id

JOIN plano\_tipo pt ON a.plano = pt.nome

GROUP BY mes, a.plano

ORDER BY mes, a.plano;

### Classifica assinantes como Ativo, Expirado ou Inativo, com base no status e data.

SELECT

u.nome,

u.email,

CASE

WHEN a.ativa = true AND CURRENT\_DATE <= a.data\_fim THEN 'Ativo'

WHEN a.ativa = false AND CURRENT\_DATE > a.data\_fim THEN 'Expirado'

ELSE 'Inativo'

END AS status\_assinatura

FROM usuario u

JOIN assinante s ON u.id = s.id

LEFT JOIN assinatura a ON s.id = a.assinante\_id;

### Calcula o valor médio pago (ticket médio) no último mês registrado, separado por forma de pagamento.

SELECT

forma,

ROUND(AVG(valor), 2) AS ticket\_medio

FROM pagamento

WHERE DATE\_TRUNC('month', data\_pagamento) = (

SELECT DATE\_TRUNC('month', MAX(data\_pagamento)) FROM pagamento

)

GROUP BY forma;

* Lista os top 5 assinantes que mais pagaram, útil para fidelização ou benefícios VIP.

SELECT

u.nome,

u.email,

SUM(p.valor) AS total\_pago

FROM usuario u

JOIN assinante s ON u.id = s.id

JOIN assinatura a ON s.id = a.assinante\_id

JOIN pagamento p ON a.id = p.assinatura\_id

GROUP BY u.nome, u.email

ORDER BY total\_pago DESC

LIMIT 5;

### Percentual de vídeos gratuitos por playlist..

SELECT

pl.titulo AS playlist,

COUNT(\*) AS total\_videos,

SUM(CASE WHEN v.gratuito = true THEN 1 ELSE 0 END) AS total\_gratuitos,

ROUND(

100.0 \* SUM(CASE WHEN v.gratuito = true THEN 1 ELSE 0 END) / COUNT(\*),

2

) AS percentual\_gratuito

FROM playlist pl

JOIN playlist\_videos pv ON pl.id = pv.playlist\_id

JOIN video v ON pv.video\_id = v.id

GROUP BY pl.titulo

ORDER BY percentual\_gratuito DESC;

Previsão de relatórios (5) que serão necessários, basta colocar o SQL necessário para obter o relatório do BD como uma breve descrição do que cada SQL retorna. Join deve ser usado em pelo menos 2 relatórios e 1 deles deve recuperar dados de pelo menos 3 tabelas. Consultas muito simples não serão consideradas, use case when ou consultas aninhadas e outros itens;

● Planejamento de backup para o BD, basta descrever como você planejou o backup e escrever o script (não apenas o comando de backup) para a realização do mesmo. O que script é que deve ser agendado para execução contínua;

● Criar trigger, function e procedure. Pensar em aplicações interessantes considerando o contexto do projeto, neste item a criatividade também conta pontos;