

**IMPLEMENTAÇÃO DE BANCO DE DADOS
IMDB (INTERNET MOVIE DATABASE)**



Arthur Ferraz

Marcelo Vieitas

Disciplina: Banco de Dados II

Prof. Luiz André

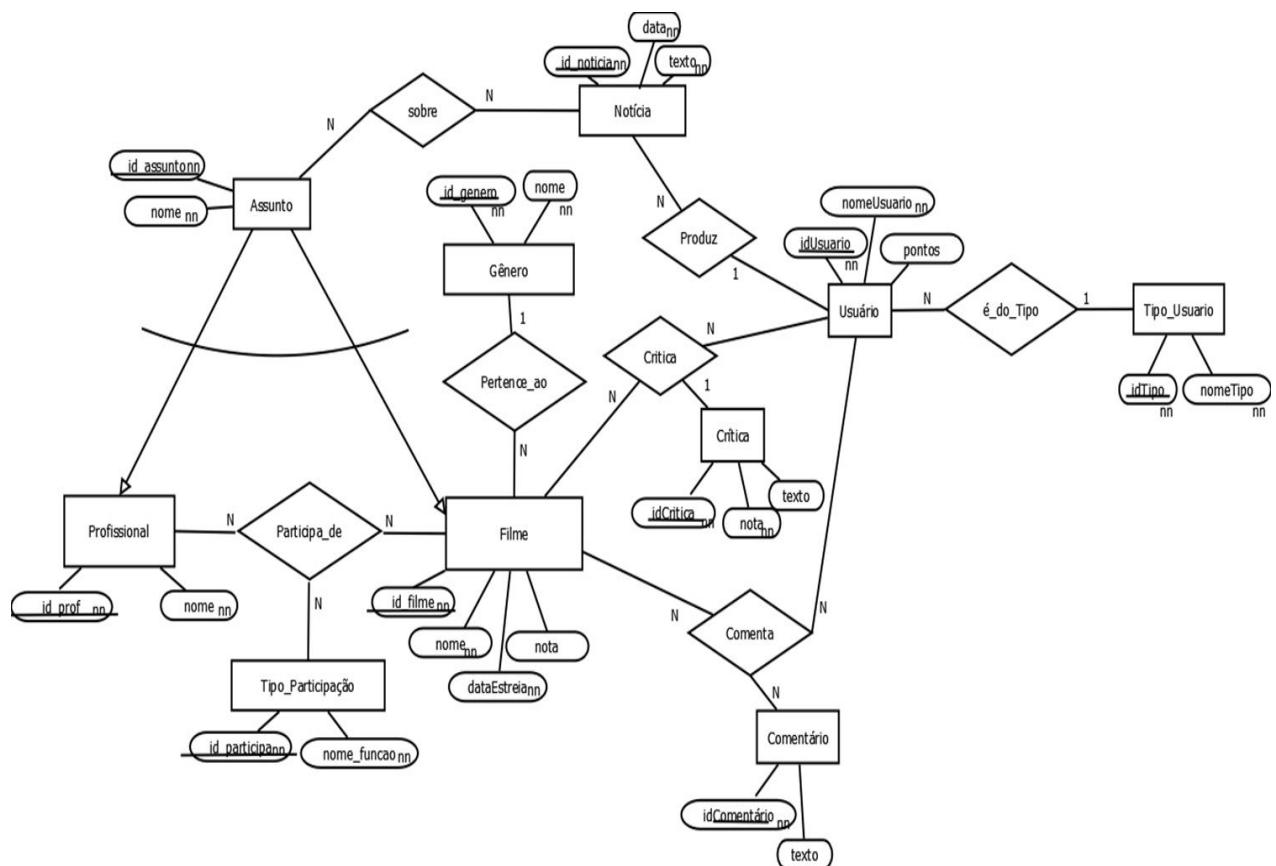
Descrição da aplicação

Trata-se de uma aplicação que tem como o seu objetivo principal proporcionar aos usuários a escrita de críticas com notas para filmes. Assim, também mostrar a média de nota de cada filme com a possibilidade de leitura das críticas.

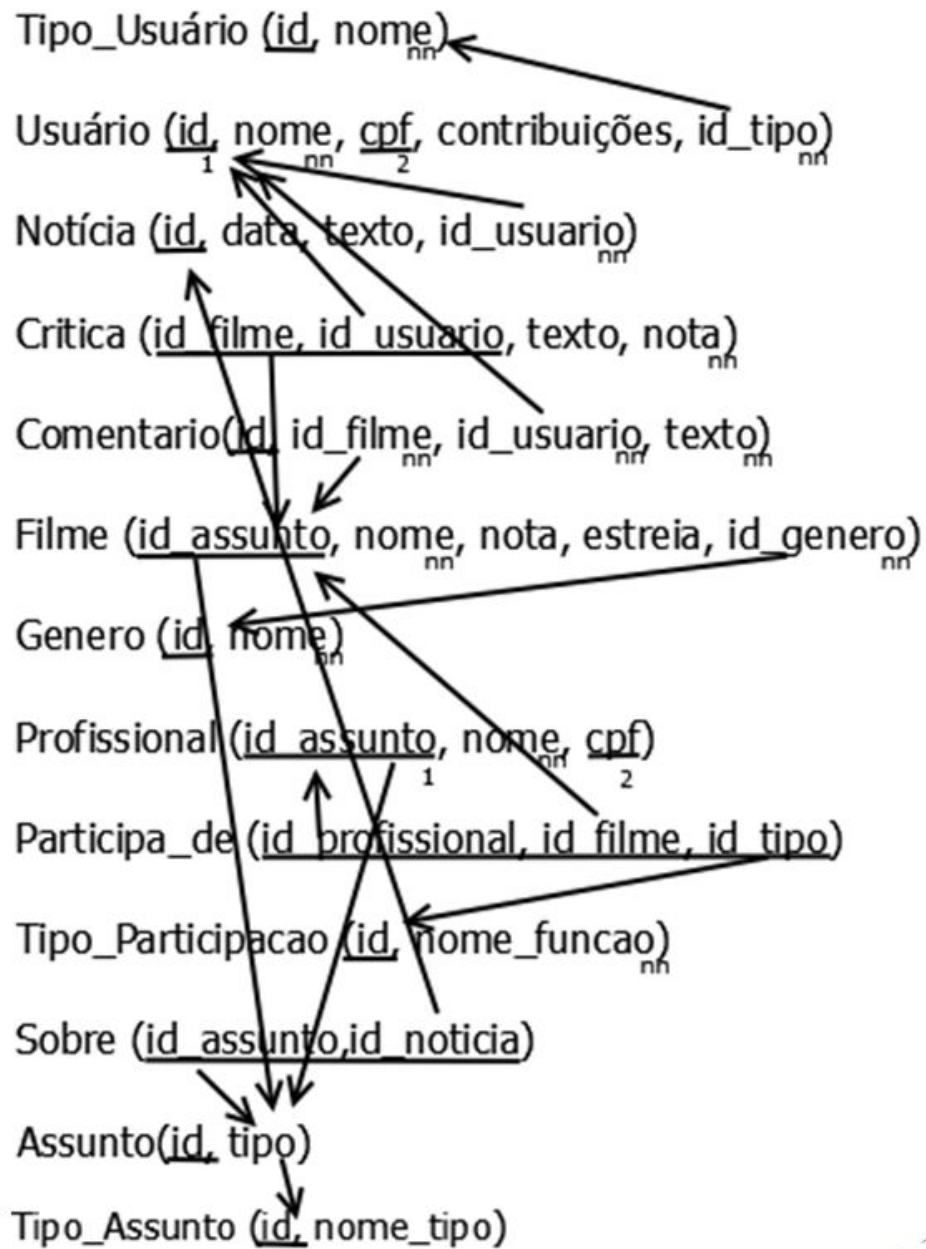
Além disso, tem como possibilidade de escrita de comentários, assim como notícias sobre determinada obra, elencar profissionais do cinema a seus filmes.

Seu funcionamento é comparado ao de um catálogo de filmes, onde se escolhe determinado filme e sobre ele existem informações, como data de estréia, nota de críticas, profissionais elencados...

Modelo relacional



Modelo lógico



Regras de Negócio

A aplicação tem como regras de negócio:

1. O usuário padrão não pode fazer uma crítica de um filme que ainda não tenha estreado.
2. As notas dos filmes devem ser atualizadas na criação de uma crítica nova por algum usuário.

3. O número de contribuições dos usuários deve ser incrementado sempre que este criar uma notícia ou crítica.
4. Deve-se manter a consistência dos tipos de assunto.
5. Uma notícia só pode conter um profissional como assunto caso este participe de pelo menos um filme como assunto da mesma notícia.

Criação do banco de dados

```
CREATE TABLE imdb.TIPO_USUARIO(  
  ID INTEGER PRIMARY KEY,  
  NOME VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE imdb.USUARIO(  
  ID INTEGER PRIMARY KEY,  
  NOME VARCHAR(200) NOT NULL,  
  CPF VARCHAR(14) NOT NULL,  
  CONTRIBUICOES INTEGER DEFAULT 0,  
  ID_TIPO INTEGER NOT NULL  
);
```

```
CREATE TABLE imdb.NOTICIA(  
  ID INTEGER PRIMARY KEY,  
  DATA_LANCAMENTO DATE NOT NULL,  
  TEXTO VARCHAR(1000) NOT NULL,  
  ID_USUARIO INTEGER NOT NULL  
);
```

```
CREATE TABLE imdb.CRITICA(  
  ID_FILME INTEGER NOT NULL,  
  ID_USUARIO INTEGER NOT NULL,  
  TEXTO VARCHAR(500),  
  NOTA FLOAT NOT NULL  
);
```

```
CREATE TABLE imdb.COMENTARIO(  
  ID INTEGER PRIMARY KEY,  
  ID_FILME INTEGER NOT NULL,  
  ID_USUARIO INTEGER NOT NULL,  
  TEXTO VARCHAR(500) NOT NULL  
);
```

);

```
CREATE TABLE imdb.FILME(  
    ID_ASSUNTO INTEGER PRIMARY KEY,  
    NOME VARCHAR(100) NOT NULL,  
    NOTA DECIMAL,  
    DATA_ESTREIA DATE,  
    ID_GENERO INTEGER NOT NULL  
);
```

```
CREATE TABLE imdb.GENERO(  
    ID INTEGER PRIMARY KEY,  
    NOME VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE imdb.PROFISSIONAL(  
    ID_ASSUNTO INTEGER PRIMARY KEY,  
    NOME VARCHAR(100) NOT NULL,  
    CPF VARCHAR(14) NOT NULL  
);
```

```
CREATE TABLE imdb.TIPO_PARTICIPACAO(  
    ID INTEGER PRIMARY KEY,  
    NOME_FUNCAO VARCHAR(200) NOT NULL  
);
```

```
CREATE TABLE imdb.PARTICIPA_DE(  
    ID_FILME INTEGER NOT NULL,  
    ID_PROFISSIONAL INTEGER NOT NULL,  
    ID_TIPO_PARTICIPACAO INTEGER NOT NULL  
);
```

```
CREATE TABLE imdb.SOBRE(  
    ID_ASSUNTO INTEGER,  
    ID_NOTICIA INTEGER  
);
```

```
CREATE TABLE imdb.ASSUNTO(  
    ID INTEGER PRIMARY KEY,  
    ID_TIPO INTEGER NOT NULL
```

);

```
CREATE TABLE imdb.TIPO_ASSUNTO(  
  ID INTEGER PRIMARY KEY,  
  TIPO VARCHAR(200) NOT NULL  
);
```

--ALTER TABLES

--ASSUNTO

```
ALTER TABLE imdb.ASSUNTO  
ADD CONSTRAINT FK_TIPO_ASSUNTO  
FOREIGN KEY (ID_TIPO) REFERENCES imdb.TIPO_ASSUNTO(ID);
```

--USUARIO

```
ALTER TABLE imdb.USUARIO  
ADD CONSTRAINT FK_TIPO_USUARIO  
FOREIGN KEY (ID_TIPO) REFERENCES imdb.TIPO_USUARIO(ID);
```

```
ALTER TABLE imdb.USUARIO  
ADD CONSTRAINT UN_CPF  
UNIQUE(CPF);
```

--NOTICIA

```
ALTER TABLE imdb.NOTICIA  
ADD CONSTRAINT FK_NOTICIA  
FOREIGN KEY (ID_USUARIO) REFERENCES imdb.USUARIO(ID);
```

--CRITICA

```
ALTER TABLE imdb.CRITICA  
ADD CONSTRAINT PK_CRITICA  
PRIMARY KEY (ID_FILME,ID_USUARIO);
```

```
ALTER TABLE imdb.CRITICA  
ADD CONSTRAINT FK_FILME  
FOREIGN KEY (ID_FILME) REFERENCES imdb.FILME(ID_ASSUNTO);
```

```
ALTER TABLE imdb.CRITICA  
ADD CONSTRAINT FK_USUARIO  
FOREIGN KEY (ID_USUARIO) REFERENCES imdb.USUARIO(ID);
```

--COMENTARIO

ALTER TABLE imdb.COMENTARIO

ADD CONSTRAINT FK_FILME

FOREIGN KEY (ID_FILME) REFERENCES imdb.FILME(ID_ASSUNTO);

ALTER TABLE imdb.COMENTARIO

ADD CONSTRAINT FK_USUARIO

FOREIGN KEY (ID_USUARIO) REFERENCES imdb.USUARIO(ID);

--FILME

ALTER TABLE imdb.FILME

ADD CONSTRAINT FK_ASSUNTO

FOREIGN KEY (ID_ASSUNTO) REFERENCES imdb.ASSUNTO(ID);

ALTER TABLE imdb.FILME

ADD CONSTRAINT FK_GENERO

FOREIGN KEY (ID_GENERO) REFERENCES imdb.GENERO(ID);

--PROFISSIONAL

ALTER TABLE imdb.PROFISSIONAL

ADD CONSTRAINT UN_CPF_PROFISSIONAL

UNIQUE(CPF);

ALTER TABLE imdb.PROFISSIONAL

ADD CONSTRAINT FK_ASSUNTO

FOREIGN KEY (ID_ASSUNTO) REFERENCES imdb.ASSUNTO(ID);

--PARTICIPA_DE

ALTER TABLE imdb.PARTICIPA_DE

ADD CONSTRAINT PK_PARTICIPA_DE

PRIMARY KEY (ID_FILME,ID_PROFISSIONAL,ID_TIPO_PARTICIPACAO);

ALTER TABLE imdb.PARTICIPA_DE

ADD CONSTRAINT FK_FILME

FOREIGN KEY (ID_FILME) REFERENCES imdb.FILME(ID_ASSUNTO);

ALTER TABLE imdb.PARTICIPA_DE

ADD CONSTRAINT FK_PROFISSIONAL

FOREIGN KEY (ID_PROFISSIONAL) REFERENCES

imdb.PROFISSIONAL(ID_ASSUNTO);

```
ALTER TABLE imdb.PARTICIPA_DE  
ADD CONSTRAINT FK_TIPO_PARTICIPACAO  
FOREIGN KEY (ID_TIPO_PARTICIPACAO) REFERENCES  
imdb.TIPO_PARTICIPACAO(ID);
```

```
--SOBRE  
ALTER TABLE imdb.SOBRE  
ADD CONSTRAINT PK_SOBRE  
PRIMARY KEY (ID_ASSUNTO, ID_NOTICIA);
```

```
ALTER TABLE imdb.SOBRE  
ADD CONSTRAINT FK_ASSUNTO  
FOREIGN KEY (ID_ASSUNTO) REFERENCES imdb.ASSUNTO(ID);
```

```
ALTER TABLE imdb.SOBRE  
ADD CONSTRAINT FK_NOTICIA  
FOREIGN KEY (ID_NOTICIA) REFERENCES imdb.NOTICIA(ID);
```

Relatórios

```
--QUAL GÊNERO DE FILME TEM MAIS CRÍTICAS?  
SELECT t2.NOME as GENERO, COUNT(TEXTOS) as CRITICAS  
FROM imdb.FILME t1  
INNER JOIN imdb.GENERO t2 ON t1.ID_GENERO = t2.ID  
LEFT JOIN imdb.CRITICA t3 ON t1.ID_ASSUNTO = t3.ID_FILME  
GROUP BY t2.NOME;
```

```
--O filme que tem mais críticas é o que tem mais comentários?  
SELECT t1.NOME as FILME, COUNT(t3.TEXTOS) as CRITICAS, COUNT(t2.TEXTOS) as  
COMENTARIOS  
FROM imdb.FILME t1  
LEFT JOIN imdb.COMENTARIO t2 ON t1.ID_ASSUNTO = t2.ID_FILME  
LEFT JOIN imdb.CRITICA t3 ON t1.ID_ASSUNTO = t3.ID_FILME  
GROUP BY t1.NOME;
```

```
--O filme que tem mais críticas é o que tem mais notícias?  
SELECT t1.NOME as FILME, COUNT(t5.TEXTOS) as CRITICAS, COUNT(t4.ID) as  
NOTICIAS  
FROM imdb.FILME t1
```



```
INNER JOIN imdb.ASSUNTO t2 ON t1.ID_ASSUNTO = t2.ID
LEFT JOIN imdb.SOBRE t3 ON t2.ID = t3.ID_ASSUNTO
LEFT JOIN imdb.NOTICIA t4 ON t3.ID_NOTICIA = t4.ID
LEFT JOIN imdb.CRITICA t5 ON t1.ID_ASSUNTO = t5.ID_FILME
GROUP BY t1.NOME;
```