



UNIVERSIDADE
ESTADUAL DE LONDRINA

MALCON MATHEUS GARCIA MARCELINO

GERALDO ANTONIO SILVA NETO

BANCO DE DADOS

LONDRINA - PR
2025

SISTEMA DE AVALIAÇÃO DE JOGOS

Descrição do sistema: O sistema que desejamos desenvolver tem o propósito de coletar, armazenar e apresentar avaliações de diferentes títulos, permitindo análises sobre história, trilha sonora, jogabilidade, visual, balanceamento e dificuldade. Além de um comentário escrito e uma recomendação positiva ou negativa.

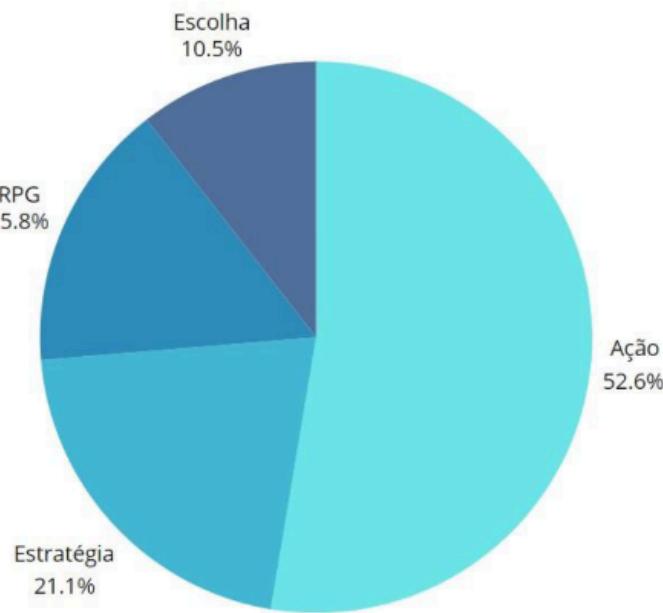
Os usuários terão um nome de usuário, e-mail e senha para autenticação e submissão de avaliações.

Cada jogo terá um título, desenvolvedora, data de lançamento, gênero, plataforma.

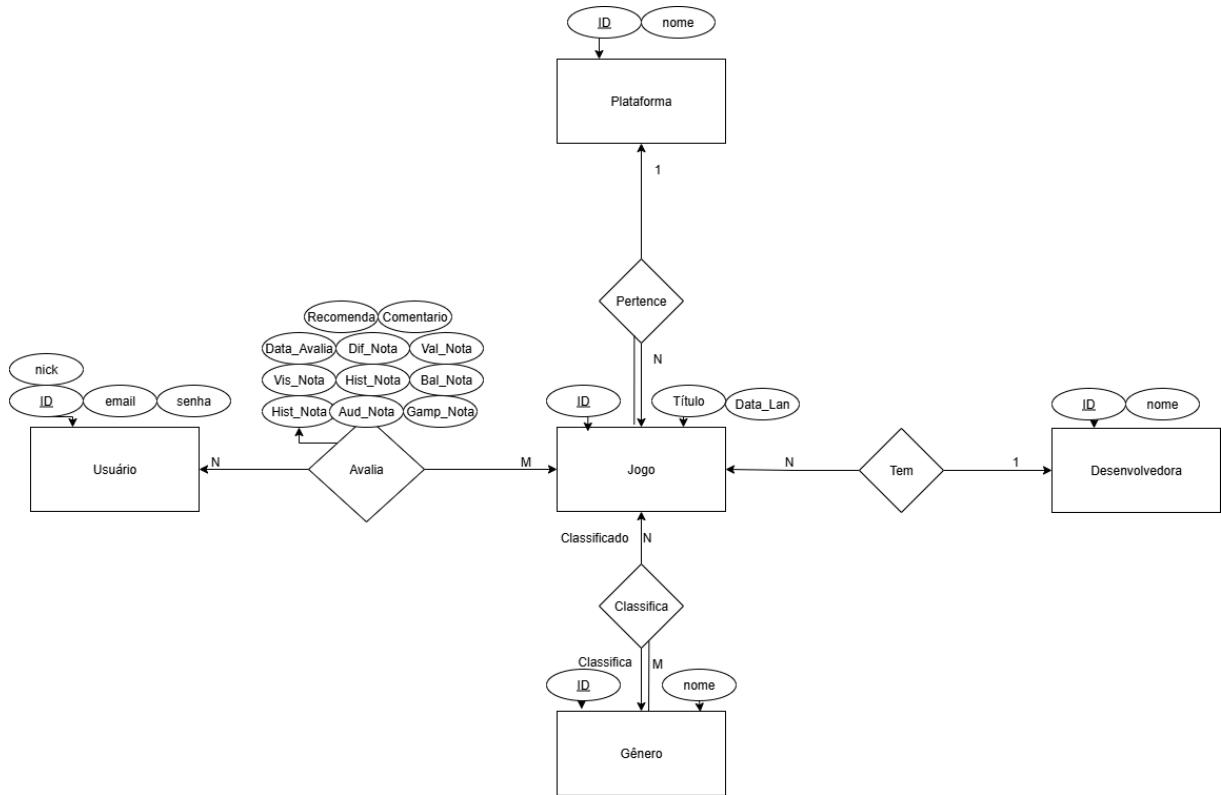
Fluxo de uso:

1. Login/Autenticação: Usuário faz login no sistema.
2. Seleção de Jogos: Usuário seleciona algum jogo para avaliar.
3. Avaliação do jogo com notas de 1 a 10 para diferentes categorias, como jogabilidade, história entre outras.

Exemplo de relatório que almejamos alcançar: gráfico com os jogos mais bem avaliados separados por gênero



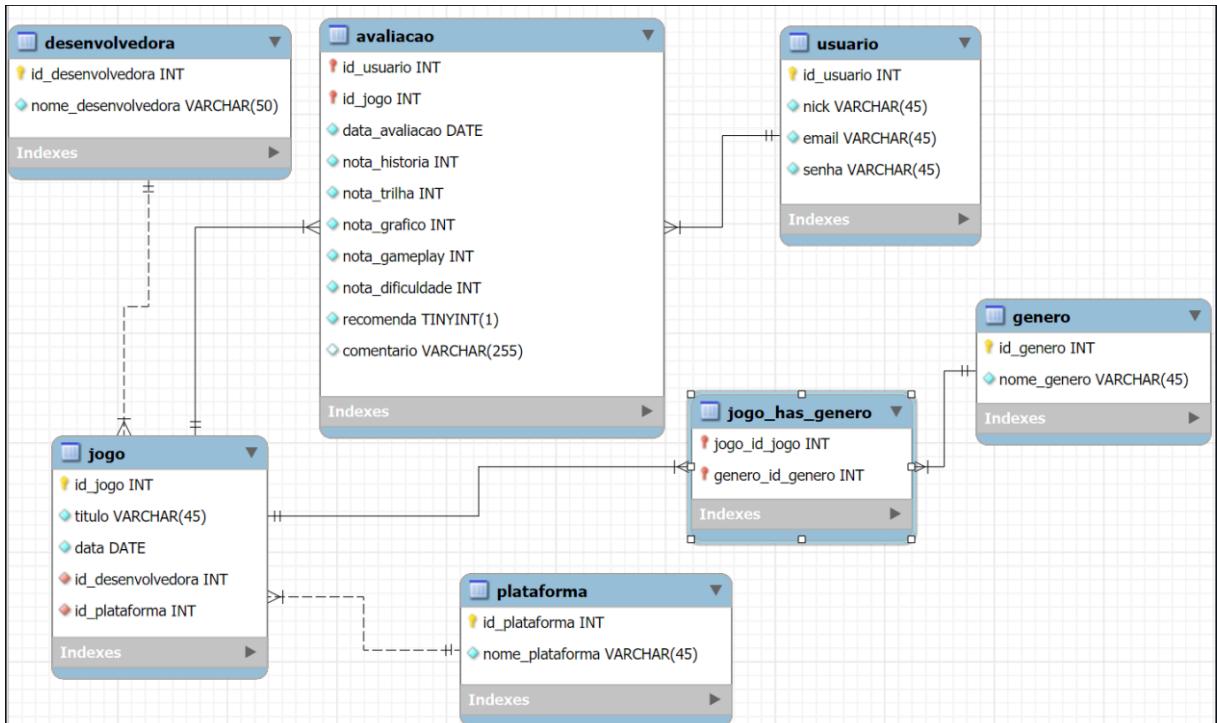
Modelo Entidade e Relacionamento



Modelo Relacional

Jogo(id, id_desenvolvedora, id_plataforma, titulo, data_lancamento)
 Desenvolvedora(id, nome)
 Plataforma(id, nome)
 Genero(id, nome)
 Usuario(id, nick, email, senha)
 Avaliacao(id_usuario^{usuário}, id_jogo^{jogo}, nota_avaliacao, nota_hist, nota_trabalho, nota_gamification, nota_dificuldade, recomendado, comentario)
 Jogo_has_genero(jogo_id^{jogo}, genero_id^{genero})

Tabelas



Tabelas SQL

```
-- Cria schema "t1"
CREATE SCHEMA IF NOT EXISTS t1;

-- --- Tabela desenvolvedora ---
CREATE TABLE IF NOT EXISTS t1.desenvolvedora
(
    -- SERIAL é o tipo de dado para auto-incremento no PostgreSQL (equivale a INT NOT
    -- NULL AUTO_INCREMENT)
    id_desenvolvedora SERIAL PRIMARY KEY,
    nome_desenvolvedora VARCHAR(50) NOT NULL,
    -- Adiciona uma restrição de unicidade para o nome da desenvolvedora
    CONSTRAINT nome_desenvolvedora_unique UNIQUE (nome_desenvolvedora)
);

-- --- Tabela plataforma ---
CREATE TABLE IF NOT EXISTS t1.plataforma
(
    id_plataforma SERIAL PRIMARY KEY,
    nome_plataforma VARCHAR(45) NOT NULL,
    CONSTRAINT nome_plataforma_unique UNIQUE (nome_plataforma)
);

-- --- Tabela genero ---
CREATE TABLE IF NOT EXISTS t1.genero
(
    id_genero SERIAL PRIMARY KEY,
    nome_genero VARCHAR(45) NOT NULL
);

-- --- Tabela jogo ---
CREATE TABLE IF NOT EXISTS t1.jogo
(
    id_jogo SERIAL PRIMARY KEY,
    titulo VARCHAR(45) NOT NULL,
    data_lancamento DATE NOT NULL,
    id_desenvolvedora INT NOT NULL,
    id_plataforma INT NOT NULL,

    -- Define a chave estrangeira para a tabela "desenvolvedora"
    CONSTRAINT fk_jogo_desenvolvedora
        FOREIGN KEY (id_desenvolvedora)
        REFERENCES t1.desenvolvedora (id_desenvolvedora)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,

    -- Define a chave estrangeira para a tabela "plataforma"
    CONSTRAINT fk_jogo_plataforma
        FOREIGN KEY (id_plataforma)

```

```
REFERENCES t1.plataforma (id_plataforma)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

-- --- Tabela usuario ---
CREATE TABLE IF NOT EXISTS t1.usuario
(
    id_usuario SERIAL PRIMARY KEY,
    nick VARCHAR(45) NOT NULL,
    email VARCHAR(45) NOT NULL,
    senha VARCHAR(45) NOT NULL,
    CONSTRAINT email_unique UNIQUE (email)
);

-- --- Tabela de relacionamento jogo_has_genero ---
-- Tabela para o relacionamento N:M entre jogo e genero
CREATE TABLE IF NOT EXISTS t1.jogo_has_genero
(
    id_jogo INT NOT NULL,
    id_genero INT NOT NULL,

    -- Define a chave primária composta
    PRIMARY KEY (id_jogo, id_genero),

    -- Define as chaves estrangeiras
    CONSTRAINT fk_jogo_has_genero_jogo
        FOREIGN KEY (id_jogo)
            REFERENCES t1.jogo (id_jogo)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_jogo_has_genero_genero
        FOREIGN KEY (id_genero)
            REFERENCES t1.genero (id_genero)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
);


```

```
-- --- Tabela avaliacao ---
-- Tabela para o relacionamento N:M entre usuario e jogo
CREATE TABLE IF NOT EXISTS t1.avaliacao
(
    id_usuario INT NOT NULL,
    id_jogo INT NOT NULL,
    data_avaliacao DATE NOT NULL,
    nota_historia INT CHECK (nota_historia BETWEEN 1 AND 10) NOT NULL,
    nota_trilha INT CHECK (nota_trilha BETWEEN 1 AND 10) NOT NULL,
    nota_grafico INT CHECK (nota_grafico BETWEEN 1 AND 10) NOT NULL,
    nota_gameplay INT CHECK (nota_gameplay BETWEEN 1 AND 10) NOT NULL,
    nota_dificuldade INT CHECK (nota_dificuldade BETWEEN 1 AND 10) NOT NULL,
    recomenda BOOLEAN NOT NULL,
    comentario VARCHAR(255) NULL,
    PRIMARY KEY (id_usuario, id_jogo),
    -- Define as chaves estrangeiras
    CONSTRAINT fk_avaliacao_usuario
        FOREIGN KEY (id_usuario)
            REFERENCES t1.usuario (id_usuario)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION,
    CONSTRAINT fk_avaliacao_jogo
        FOREIGN KEY (id_jogo)
            REFERENCES t1.jogo (id_jogo)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION
);
```