

Universidade Federal de Sergipe
Centro de Ciências Exatas e Tecnologia
Departamento de Computação

Relatório de Banco de Dados: Parte 2.1

Projeto Lógico e Implementação PostgreSQL (Steam)

Alunos: Rafael Gomes Oliveira Santos
Alícia Vitória Sousa Santos

Professor: André Britto

São Cristóvão
2026

1 Especificação e Descrição do Problema

O sistema desenvolvido visa gerir uma plataforma de distribuição digital baseada no ecossistema Steam. As principais regras de negócio implementadas são:

- **Gestão de Identidade e Finanças:** Cada usuário possui um identificador único, e-mail exclusivo e uma carteira vinculada (relacionamento 1:1) com precisão decimal para saldos.
- **Especialização de Produtos:** Utiliza-se uma hierarquia onde **Produto** é a generalização, especializada em **Jogos**, **Software** e **DLC** através de tabelas distintas com integridade referencial.
- **Normalização de Atributos:** Atributos multivvalorados (Idiomas) e compostos (Requisitos) foram normalizados para atender à 1FN.
- **Entidades Fracas:** As conquistas são dependentes existencialmente dos jogos, possuindo identificação composta.

2 Projeto Conceitual e Mapeamento Relacional

O projeto foi estruturado seguindo a metodologia de projeto de banco de dados, partindo de uma abstração de alto nível até a implementação física.

2.1 Modelo Conceitual (DER)

O Diagrama Entidade-Relacionamento (DER) utiliza a notação de Heuser para descrever a semântica do ecossistema. Destaca-se a especialização total do **Produto**, onde a cardinalidade $(0, N)$ em **Conquistas** e **Gêneros** reflete a opcionalidade desses dados para diferentes títulos. O atributo *Idiomas* é tratado como multivvalorado, prevendo sua futura implementação otimizada.

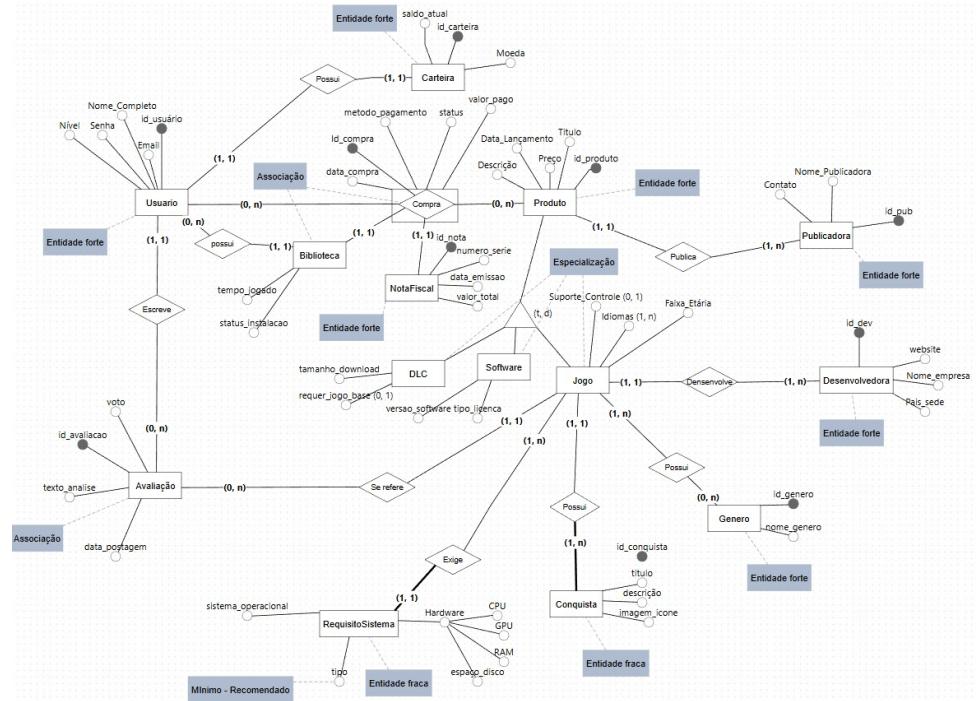


Figura 1: Modelo Conceitual - Diagrama Entidade-Relacionamento (DER)

2.2 Modelo Relacional (Lógico)

O mapeamento para o modelo relacional priorizou a manutenção das restrições de integridade. Adotou-se a estratégia de **Tabela por Subclasse** para a hierarquia de produtos. As relações Muitos-para-Muitos (N:M), como *Biblioteca* e *Gêneros*, foram resolvidas através de tabelas associativas. Foram implementadas **Chaves Naturais** para as entidades *Desenvolvedora* e *Publicadora* (baseadas em nomes), garantindo a unicidade sem a necessidade de IDs artificiais.

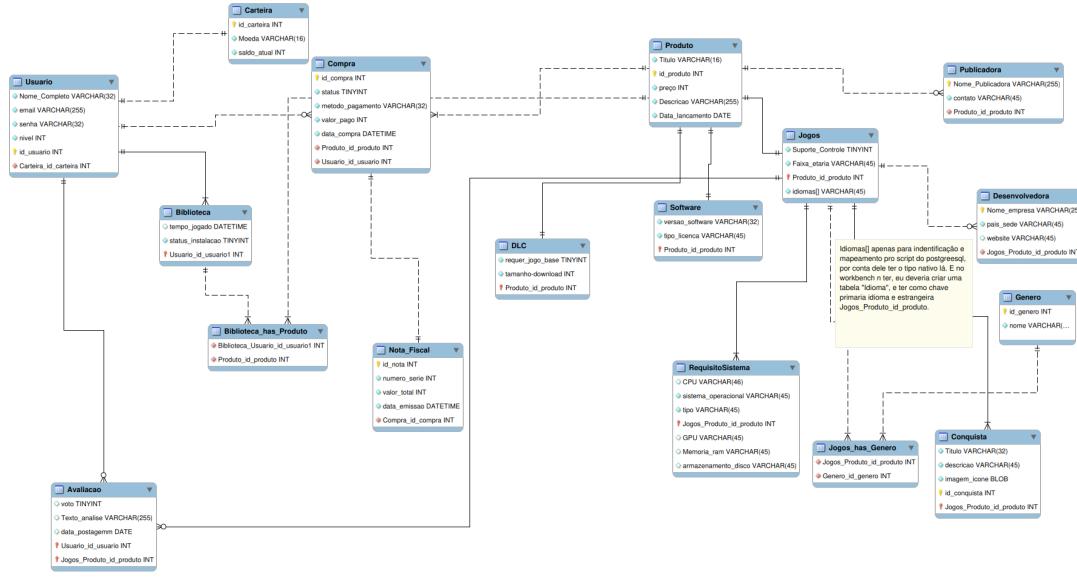


Figura 2: Modelo Relacional - Esquema Lógico/Físico

3 Implementação no PostgreSQL (Amazon RDS)

A implementação técnica no serviço gerenciado da AWS focou nas propriedades **ACID**:

- **Domínios e Tipos:** Uso de SERIAL para autoincremento, TIMESTAMP para registros temporais e NUMERIC para precisão financeira.
 - **Extensões Objeto-Relacionais:** Implementação do atributo *Idiomas* como VARCHAR[], permitindo o armazenamento de múltiplos valores em uma única coluna sem violar o desempenho em consultas de leitura.
 - **Ações Referenciais:** Uso estratégico de ON DELETE RESTRICT em tabelas de *Compra* e *Avaliação* para preservar a integridade histórica e financeira.

4 Script SQL

```
1 --  
2 -- PROJETO LOGICO: ECOSSISTEMA STEAM  
3 -- Disciplina: Banco de Dados I  
4 --  
5  
6 CREATE SCHEMA IF NOT EXISTS steam;  
7 SET search_path TO steam;  
8  
9 -- Limpeza para reinicializacao do ambiente de testes  
10 DROP TABLE IF EXISTS Jogos_has_Genero , DLC , Software , RequisitoSistema ,  
    Conquista ,  
    Nota_Fiscal , Compra , Biblioteca , Biblioteca_has_Produto , Avaliacao ,  
    Publicadora , Desenvolvedora ,  
11 Jogos , Produto , Genero , Usuario , Carteira CASCADE ;  
12  
13  
14 --  
15 -- NUCLEO: USUARIOS E FINANCEIRO  
16 --  
17  
18 CREATE TABLE IF NOT EXISTS Carteira (  
19     id_carteira SERIAL PRIMARY KEY ,  
20     Moeda VARCHAR(16) NOT NULL ,  
21     saldo_atual NUMERIC(10, 2) NOT NULL DEFAULT 0.00  
22 );  
23  
24 CREATE TABLE IF NOT EXISTS Usuario (  
25     id_usuario SERIAL PRIMARY KEY ,  
26     Nome_Completo VARCHAR(100) NOT NULL ,  
27     email VARCHAR(255) NOT NULL UNIQUE ,  
28     senha VARCHAR(32) NOT NULL ,  
29     nivel INT NOT NULL DEFAULT 1 ,  
30     id_carteira INT NOT NULL ,  
31     CONSTRAINT fk_Usuario_Carteira FOREIGN KEY (id_carteira)  
32         REFERENCES Carteira (id_carteira) ON DELETE CASCADE ON UPDATE  
            CASCADE  
33 );  
34
```

```

35 --
36 -- CATALOGO E CATEGORIZACAO
37 --
38
39 CREATE TABLE IF NOT EXISTS Genero (
40     id_genero SERIAL PRIMARY KEY,
41     nome VARCHAR(255) NOT NULL UNIQUE
42 );
43
44 CREATE TABLE IF NOT EXISTS Produto (
45     id_produto SERIAL PRIMARY KEY,
46     Titulo VARCHAR(100) NOT NULL,
47     preco NUMERIC(10, 2) NOT NULL DEFAULT 0.00,
48     Descricao TEXT NOT NULL,
49     Data_lancamento DATE NOT NULL
50 );
51
52 --
53 -- ESPECIALIZACOES (PRODUTO)
54 --
55
56 CREATE TABLE IF NOT EXISTS Jogos (
57     id_produto INT PRIMARY KEY,
58     Suporte_Controle BOOLEAN NOT NULL DEFAULT FALSE,
59     Faixa_etaria VARCHAR(45) NOT NULL,
60     idiomas VARCHAR(45)[] NOT NULL,
61     CONSTRAINT fk_Jogos_Produto FOREIGN KEY (id_produto)
62         REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
63 );
64
65 CREATE TABLE IF NOT EXISTS Software (
66     id_produto INT PRIMARY KEY,
67     versao_software VARCHAR(32) NOT NULL,
68     tipo_licenca VARCHAR(45) NOT NULL,
69     CONSTRAINT fk_Software_Produto FOREIGN KEY (id_produto)
70         REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
71 );
72
73 CREATE TABLE IF NOT EXISTS DLC (

```

```

74     id_produto INT PRIMARY KEY,
75     requer_jogo_base BOOLEAN NOT NULL DEFAULT TRUE,
76     tamanho_download VARCHAR(45),
77     CONSTRAINT fk_DLC_Produto FOREIGN KEY (id_produto)
78       REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
79   );
80
81 --
82 -- AGENTES DO MERCADO (CHAVES NATURAIS)
83 --
84
85 CREATE TABLE IF NOT EXISTS Desenvolvedora (
86     Nome_empresa VARCHAR(255) PRIMARY KEY,
87     pais_sede VARCHAR(45) NOT NULL,
88     website VARCHAR(100),
89     id_jogo INT NOT NULL,
90     CONSTRAINT fk_Desenvolvedora_Jogos FOREIGN KEY (id_jogo)
91       REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
92   );
93
94 CREATE TABLE IF NOT EXISTS Publicadora (
95     Nome_Publicadora VARCHAR(255) PRIMARY KEY,
96     contato VARCHAR(100) NOT NULL,
97     id_produto INT NOT NULL,
98     CONSTRAINT fk_Publicadora_Produto FOREIGN KEY (id_produto)
99       REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
100  );
101
102 --
103 -- INTERACAO E AUDITORIA
104 --
105
106 CREATE TABLE IF NOT EXISTS Avaliacao (
107     id_usuario INT NOT NULL,
108     id_jogo INT NOT NULL,
109     voto SMALLINT CHECK (voto >= 0 AND voto <= 5),
110     Texto_analise TEXT NULL,
111     data_postagem TIMESTAMP NULL,
112     PRIMARY KEY (id_usuario, id_jogo),

```

```

113    CONSTRAINT fk_Avaliacao_Usuario FOREIGN KEY (id_usuario)
114        REFERENCES Usuario (id_usuario) ON DELETE RESTRICT ON UPDATE CASCADE
115    ,
116    CONSTRAINT fk_Avaliacao_Jogos FOREIGN KEY (id_jogo)
117        REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
118 );
119
120 CREATE TABLE IF NOT EXISTS Biblioteca (
121     id_usuario INT PRIMARY KEY,
122     tempo_jogado TIMESTAMP NULL,
123     status_instalacao BOOLEAN DEFAULT FALSE,
124     CONSTRAINT fk_Biblioteca_Usuario FOREIGN KEY (id_usuario)
125         REFERENCES Usuario (id_usuario) ON DELETE CASCADE ON UPDATE CASCADE
126 );
127
128 CREATE TABLE IF NOT EXISTS Biblioteca_has_Produto (
129     id_usuario INT NOT NULL,
130     id_produto INT NOT NULL,
131     PRIMARY KEY (id_usuario, id_produto),
132     CONSTRAINT fk_Biblioteca_Produto_User FOREIGN KEY (id_usuario)
133         REFERENCES Biblioteca (id_usuario) ON DELETE CASCADE ON UPDATE
134         CASCADE,
135     CONSTRAINT fk_Biblioteca_Produto_Prod FOREIGN KEY (id_produto)
136         REFERENCES Produto (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
137 );
138
139 CREATE TABLE IF NOT EXISTS Compra (
140     id_compra SERIAL PRIMARY KEY,
141     status VARCHAR(20) NOT NULL,
142     metodo_pagamento VARCHAR(32) NOT NULL,
143     valor_pago NUMERIC(10, 2) NOT NULL,
144     data_compra TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
145     id_produto INT NOT NULL,
146     id_usuario INT NOT NULL,
147     CONSTRAINT fk_Compra_Produto FOREIGN KEY (id_produto)
148         REFERENCES Produto (id_produto) ON DELETE RESTRICT ON UPDATE CASCADE
149    ,
150     CONSTRAINT fk_Compra_Usuario FOREIGN KEY (id_usuario)
151         REFERENCES Usuario (id_usuario) ON DELETE RESTRICT ON UPDATE CASCADE
152 );
153
154 CREATE TABLE IF NOT EXISTS Nota_Fiscal (
155     id_nota SERIAL PRIMARY KEY,
156     numero_serie INT NOT NULL,
157     valor_total NUMERIC(10, 2) NOT NULL,
158     id_compra INT NOT NULL UNIQUE,
159     CONSTRAINT fk_Nota_Fiscal_Compra FOREIGN KEY (id_compra)

```

```

157     REFERENCES Compra (id_compra) ON DELETE CASCADE ON UPDATE CASCADE
158 );
159
160 --
161 -- DETALHES TECNICOS E PROGRESSAO
162 --
163
164 CREATE TABLE IF NOT EXISTS Jogos_has_Genero (
165     id_jogo INT NOT NULL ,
166     id_genero INT NOT NULL ,
167     PRIMARY KEY (id_jogo, id_genero),
168     CONSTRAINT fk_Jogos_Genero_Jogo FOREIGN KEY (id_jogo)
169         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE ,
170     CONSTRAINT fk_Jogos_Genero_Gen FOREIGN KEY (id_genero)
171         REFERENCES Genero (id_genero) ON DELETE RESTRICT ON UPDATE CASCADE
172 );
173
174 CREATE TABLE IF NOT EXISTS Conquista (
175     id_conquista INT NOT NULL ,
176     id_jogo INT NOT NULL ,
177     Titulo VARCHAR(100) NOT NULL ,
178     descricao TEXT NOT NULL ,
179     imagem_icone BYTEA NOT NULL ,
180     PRIMARY KEY (id_conquista, id_jogo),
181     CONSTRAINT fk_Conquista_Jogos FOREIGN KEY (id_jogo)
182         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
183 );
184
185 CREATE TABLE IF NOT EXISTS RequisitoSistema (
186     id_requisito SERIAL PRIMARY KEY,
187     CPU VARCHAR(100),
188     GPU VARCHAR(100),
189     Memoria_ram VARCHAR(45),
190     armazenamento_disco VARCHAR(45),
191     sistema_operacional VARCHAR(45) NOT NULL ,
192     tipo VARCHAR(45) NOT NULL CHECK (tipo IN ('Minimo', 'Recomendado')),
193     id_jogo INT NOT NULL ,
194     CONSTRAINT fk_Requisito_Jogos FOREIGN KEY (id_jogo)
195         REFERENCES Jogos (id_produto) ON DELETE CASCADE ON UPDATE CASCADE
196 );
197
198 --

```

```

199 -- PERFORMANCE E ACESSO
200 --
201
202 CREATE INDEX IF NOT EXISTS idx_usuario_email ON Usuario(email);
203 CREATE INDEX IF NOT EXISTS idx_compra_data ON Compra(data_compra);
204
205 DO $$*
206 BEGIN
207   IF NOT EXISTS (SELECT FROM pg_catalog.pg_roles WHERE rolname = 'professor') THEN
208     CREATE USER professor WITH PASSWORD 'professor';
209   END IF;
210 END $$;
211
212 GRANT USAGE ON SCHEMA steam TO professor;
213 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA steam TO professor;
214 GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA steam TO professor;
215 ALTER USER professor SET search_path TO steam, public;

```

5 Instância Amazon RDS

O banco de dados encontra-se ativo e disponível para testes docentes. O Security Group foi configurado para permitir tráfego de entrada na porta 5432.

- **Endpoint:** steamdb.cximu3gskw3g.us-east-1.rds.amazonaws.com
- **Credenciais:** Usuário professor | Senha professor.