



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO**

**DISCIPLINA: BANCO DE DADOS**

**DOCENTE: BRUNO BORGES DA SILVA**

**LOCADORA DE JOGOS**

**Componentes :** Maria Francisca Pessoa de Queiroz

Gabrielle Naiare Pereira de Oliveira

Jennefhy Saiury de Lima Pontes

Pablo Ryan Lopes Vieira

Pedro Victor Soares Gonçalo

**Pau dos Ferros, RN**

**2025**

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>2</b>
1.1 REQUISITOS DETALHADOS.....	2
<b>2 MODELAGEM DE DADOS.....</b>	<b>2</b>
2.1 MODELO CONCEITUAL.....	2
2.2 MODELO LÓGICO.....	2
2.3 MODELO FÍSICO (SCRIPT SQL).....	2
2.4 PROCESSO DE NORMALIZAÇÃO.....	3
<b>3 DICIONÁRIO DE DADOS.....</b>	<b>3</b>
<b>4 IMPLEMENTAÇÃO E RECURSOS AVANÇADOS.....</b>	<b>4</b>
4.1 OPERAÇÕES BÁSICOS SQL.....	4
4.2 VIEWS.....	4
4.3 GATILHOS (TRIGGERS).....	4
4.4 FUNÇÕES.....	4
<b>5 PLANO DE TESTES.....</b>	<b>4</b>
5.1 TESTES DE INSERÇÃO.....	4
5.2 TESTES DE REMOÇÃO.....	4
5.3 TESTES DE LISTAGEM (SELECT).....	4
5.4 TESTES DE ALTERAÇÃO (UPDATE).....	4
5.5 TESTES DE VIEWS.....	4
5.6 TESTES DE GATILHOS (TRIGGERS).....	4
5.7 TESTE DE FUNÇÕES.....	4
<b>6 CONCLUSÃO.....</b>	<b>4</b>
<b>7 REFERÊNCIAS (Se houver).....</b>	<b>4</b>

## 1 INTRODUÇÃO

O tema do banco de dados é um sistema de locadora de jogos, responsável por gerenciar informações relacionadas a clientes, jogos físicos, funcionários, alugueis e devoluções. O objetivo principal é organizar o fluxo de empréstimo de jogos, garantindo controle sobre estoque, prazos e registros de transações.

O sistema foi modelado para atender às necessidades de uma locadora que oferece diversos jogos físicos, permitindo que clientes realizem locações mediante cadastro e acompanhamento por funcionários.

### **Funcionalidades Principais:**

O banco de dados suporta as seguintes funcionalidades essenciais:

- Cadastro e gerenciamento de clientes
- Cadastro de funcionários
- Registro de jogos físicos e seu estoque
- Controle de alugueis e devoluções
- Atualização automática do estoque através de triggers
- Consultas gerais (clientes cadastrados, jogos mais alugados, alugueis ativos, etc.)
- Controle de multa por atraso
- Rastreabilidade completa das transações realizadas

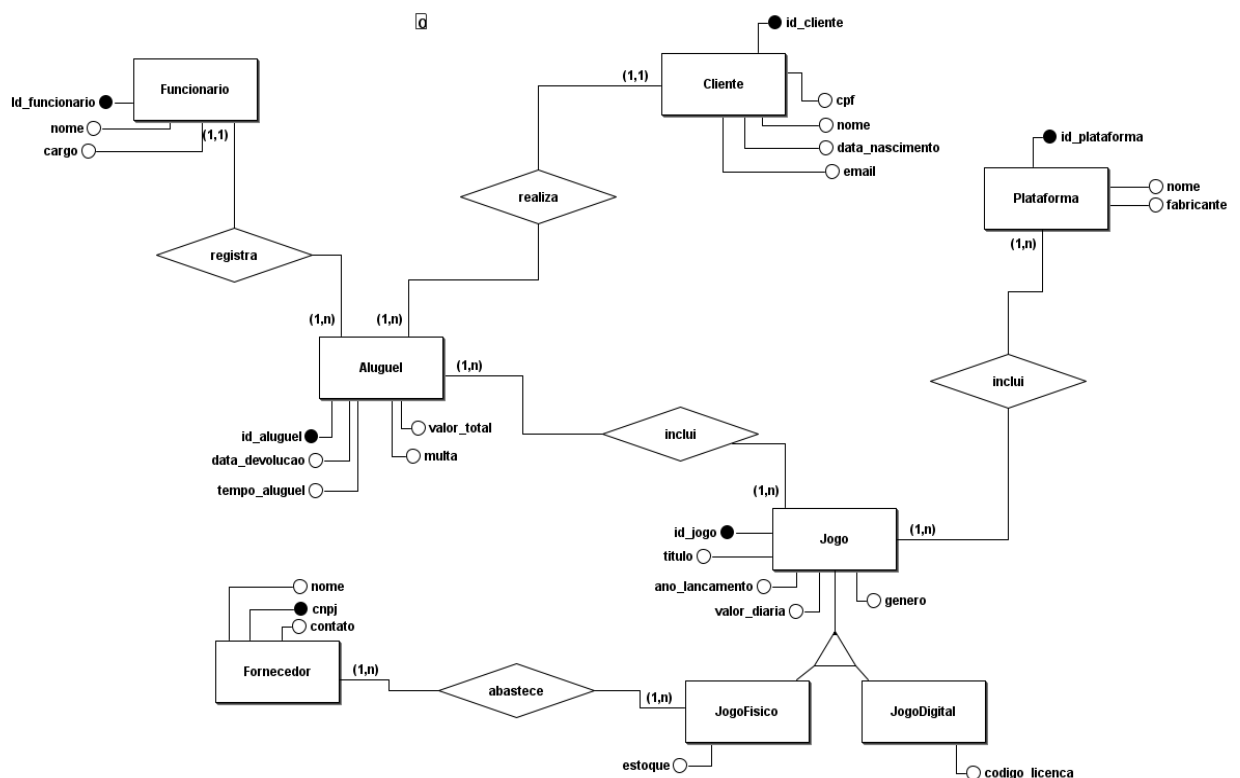
### 1.1 REQUISITOS DETALHADOS

1. O sistema deve permitir o cadastro, edição e exclusão de clientes.
2. O sistema deve permitir o cadastro de funcionários responsáveis pelos atendimentos.
3. O sistema deve permitir o registro de jogos físicos, incluindo título, gênero, ano e quantidade em estoque.

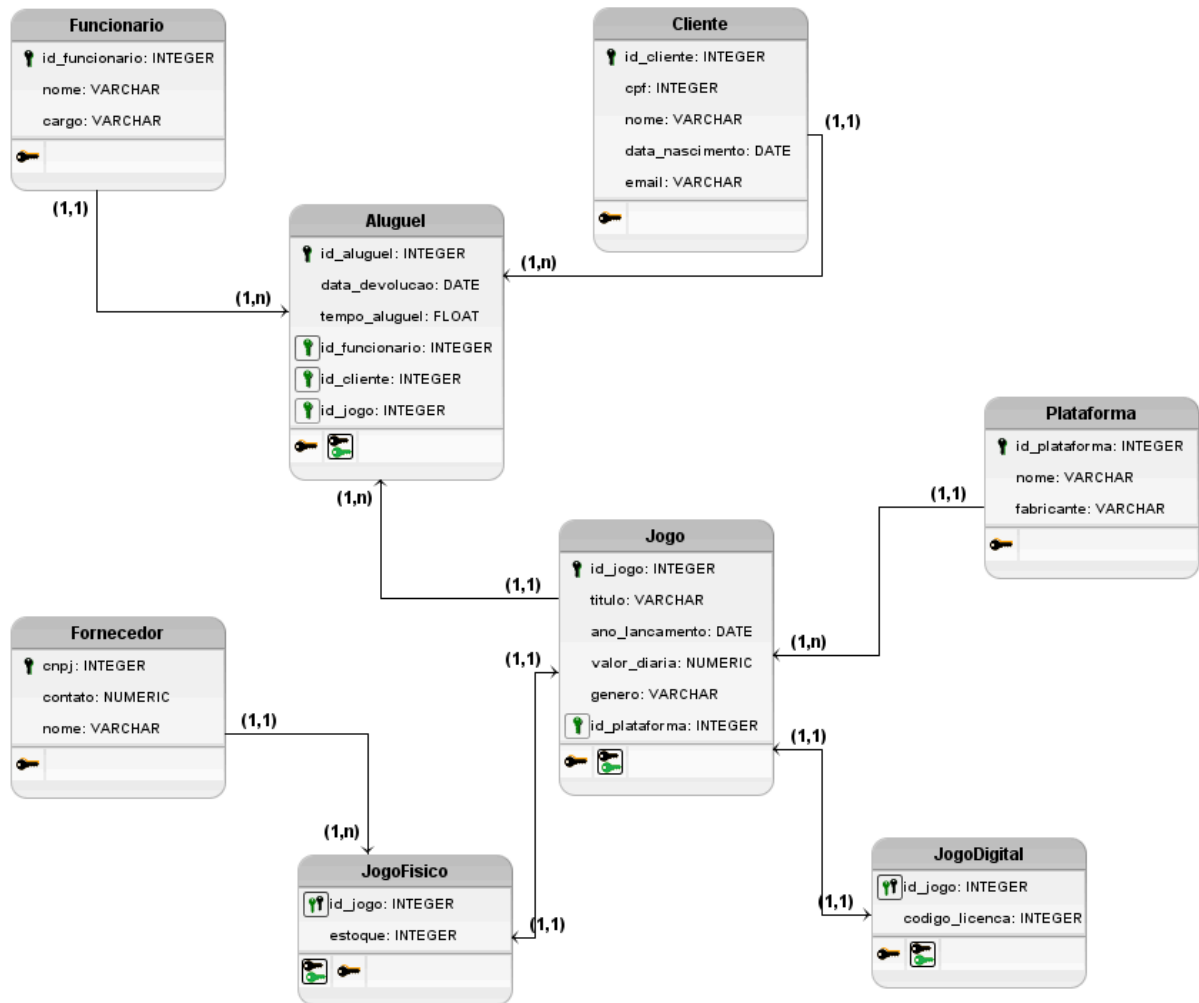
4. O sistema deve permitir registrar um aluguel, vinculando cliente, jogo e funcionário.
5. O sistema deve atualizar automaticamente o estoque ao realizar um aluguel (estoque – 1).
6. O sistema deve atualizar automaticamente o estoque ao registrar uma devolução (estoque + 1).
7. O sistema deve permitir consultar todos os alugueis ativos.
8. O sistema deve permitir consultar o histórico de alugueis de um cliente.
9. O sistema deve permitir registrar a devolução, incluindo cálculo de multa por atraso quando aplicável.
10. O sistema deve impedir aluguel caso o jogo esteja com estoque esgotado.
11. O sistema deve permitir consultar jogos disponíveis e filtros por gênero.
12. O sistema deve manter integridade referencial, impedindo exclusão de registros quando dependências existirem.
13. O sistema deve registrar automaticamente data e hora de operações principais, como aluguel e devolução.

## 2 MODELAGEM DE DADOS

### 2.1 MODELO CONCEITUAL



## 2.2 MODELO LÓGICO



## 2.3 MODELO FÍSICO (SCRIPT SQL)

No modelo físico, há a apresentação da Locadora de Jogos no Script SQL, utilizando como ferramenta o **PostgreSQL**.

- Criação do banco de dados `locadora_jogos` - **CREATE DATABASE**

```
-- Database: locadora_jogos
```

```
-- DROP DATABASE IF EXISTS locadora_jogos;
```

```
CREATE DATABASE locadora_jogos
```

```
WITH
OWNER = postgres
ENCODING = 'UTF8'
LC_COLLATE = 'Portuguese_Brazil.1252'
LC_CTYPE = 'Portuguese_Brazil.1252'
LOCALE_PROVIDER = 'libc'
TABLESPACE = pg_default
CONNECTION LIMIT = -1
IS_TEMPLATE = False;
```

- Criação da Tabela Funcionario - **CREATE TABLE**

```
CREATE TABLE Funcionario (
    id_funcionario SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    cargo VARCHAR(50) NOT NULL
);
```

- Criação da Tabela Cliente - **CREATE TABLE**

```
CREATE TABLE Cliente (
    id_cliente SERIAL PRIMARY KEY,
    cpf VARCHAR(14) UNIQUE NOT NULL,
    nome VARCHAR(100) NOT NULL,
    data_nascimento DATE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL
);
```

- Criação da Tabela Plataforma - **CREATE TABLE**

```
CREATE TABLE Plataforma (
    id_plataforma SERIAL PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    fabricante VARCHAR(50)
);
```

- Criação da Tabela Jogo - **CREATE TABLE**

```
CREATE TABLE Jogo (  
    id_jogo SERIAL PRIMARY KEY,  
    titulo VARCHAR(200) NOT NULL,  
    ano_lancamento INTEGER,  
    valor_diaria NUMERIC(10, 2) NOT NULL,  
    genero VARCHAR(50)  
);
```

- Criação da Tabela JogoFisico - **CREATE TABLE**

```
CREATE TABLE JogoFisico (  
    id_jogo INTEGER PRIMARY KEY REFERENCES Jogo(id_jogo),  
    estoque INTEGER NOT NULL CHECK (estoque >= 0)  
);
```

- Criação da Tabela JogoDigital - **CREATE TABLE**

```
CREATE TABLE JogoDigital (  
    id_jogo INTEGER PRIMARY KEY REFERENCES Jogo(id_jogo),  
    codigo_licenca VARCHAR(50) UNIQUE NOT NULL  
);
```

- Criação da Tabela Fornecedor - **CREATE TABLE**

```
CREATE TABLE Fornecedor (  
    cnpj VARCHAR(18) PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    contato VARCHAR(100)  
);
```

- Criação da Tabela Aluguel - **CREATE TABLE**

```
CREATE TABLE Aluguel (  
    id_aluguel SERIAL PRIMARY KEY,  
    data_aluguel TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT  
NULL,  
    data_devolucao TIMESTAMP,
```

```

    tempo_aluguel INTEGER,
    valor_total NUMERIC(10, 2),
    multa NUMERIC(10, 2) DEFAULT 0.00,
    id_funcionario      INTEGER      NOT      NULL      REFERENCES
Funcionario(id_funcionario),
    id_cliente          INTEGER      NOT      NULL      REFERENCES
Cliente(id_cliente)
);

```

## 2.4 PROCESSO DE NORMALIZAÇÃO

### 2.4.1 MODELO TEXTUAL

#### NÃO NORMALIZADO

**Funcionario** (Id\_funcionario, nome\_funcionario, cargo\_funcionario)

**Cliente** (Id\_cliente, cpf\_cliente, nome\_cliente, data\_nascimento\_cliente, email\_cliente)

Fornecedor (cnpj\_fornecedor, contato\_fornecedor, nome\_fornecedor)

Plataforma (Id\_plataforma, nome\_plataforma, fabricante\_plataforma)

**Jogo** (Id\_jogo, titulo\_jogo, ano\_lancamento\_jogo, valor\_diaria\_jogo, genero\_jogo, estoque\_jogo\_fisico, codigo\_licenca\_jogo\_digital, Id\_plataforma, nome\_plataforma, fabricante\_plataforma, cnpj\_fornecedor, contato\_fornecedor, nome\_fornecedor)

**Aluguel** (Id\_aluguel, data\_devolucao, tempo\_aluguel, valor\_total, multa, Id\_funcionario, nome\_funcionario, cargo\_funcionario, Id\_cliente, cpf\_cliente, nome\_cliente, data\_nascimento\_cliente, email\_cliente, Id\_jogo, titulo\_jogo, ano\_lancamento\_jogo, valor\_diaria\_jogo, genero\_jogo, estoque\_jogo\_fisico, codigo\_licenca\_jogo\_digital, Id\_plataforma, nome\_plataforma, fabricante\_plataforma)

#### 1FN

**Aluguel** (Id\_aluguel[pk], data\_devolucao, tempo\_aluguel, valor\_total, multa, Id\_funcionario, Id\_cliente, Id\_jogo)

**Funcionario** (Id\_funcionario[pk], nome\_funcionario, cargo\_funcionario)

**Cliente** (Id\_cliente[pk], cpf\_cliente, nome\_cliente, data\_nascimento\_cliente, email\_cliente)

**Jogo** (Id\_jogo[pk], titulo\_jogo, ano\_lancamento\_jogo, valor\_diaria\_jogo, genero\_jogo,



estoque\_jogo\_fisico, codigo\_licenca\_jogo\_digital, Id\_plataforma, cnpj\_fornecedor) (herança: colunas opcionais para fisico/digital)

**Plataforma** (Id\_plataforma[pk], nome\_plataforma, fabricante\_plataforma)

**Fornecedor** (cnpj\_fornecedor[pk], contato\_fornecedor, nome\_fornecedor)

## 2FN

**Aluguel** (Id\_aluguel[pk], data\_devolucao, tempo\_aluguel, valor\_total, multa, Id\_funcionario, Id\_cliente, Id\_jogo)

**Funcionario** (Id\_funcionario[pk], nome\_funcionario, cargo\_funcionario)

**Cliente** (Id\_cliente[pk], cpf\_cliente, nome\_cliente, data\_nascimento\_cliente, email\_cliente)

**Jogo** (Id\_jogo[pk], titulo\_jogo, ano\_lancamento\_jogo, valor\_diaria\_jogo, genero\_jogo, Id\_plataforma)

**JogoFisico** (Id\_jogo[pk], estoque\_jogo\_fisico, cnpj\_fornecedor)

**JogoDigital** (Id\_jogo[pk], codigo\_licenca\_jogo\_digital)

**Plataforma** (Id\_plataforma[pk], nome\_plataforma, fabricante\_plataforma)

**Fornecedor** (cnpj\_fornecedor[pk], contato\_fornecedor, nome\_fornecedor)

## 3FN

**Aluguel** (Id\_aluguel[pk], data\_devolucao, tempo\_aluguel, valor\_total, multa, Id\_funcionario, Id\_cliente, Id\_jogo)

**Funcionario** (Id\_funcionario[pk], nome\_funcionario, cargo\_funcionario)

**Cliente** (Id\_cliente[pk], cpf\_cliente, nome\_cliente, data\_nascimento\_cliente, email\_cliente)

**Jogo** (Id\_jogo[pk], titulo\_jogo, ano\_lancamento\_jogo, valor\_diaria\_jogo, genero\_jogo, Id\_plataforma)

**JogoFisico** (Id\_jogo[pk], estoque\_jogo\_fisico, cnpj\_fornecedor)

**JogoDigital** (Id\_jogo[pk], codigo\_licenca\_jogo\_digital)

**Plataforma** (Id\_plataforma[pk], nome\_plataforma, fabricante\_plataforma)

**Fornecedor** (cnpj\_fornecedor[pk], contato\_fornecedor, nome\_fornecedor)

## 2.4.2 TABELAS

### 1FN

Tabela	Chave Primária	Atributos
--------	----------------	-----------

Aluguel	Id_aluguel [pk]	data_devolucao, tempo_aluguel, valor_total, multa, Id_funcionario [fk], Id_cliente [fk], Id_jogo [fk]
Funcionario	Id_funcionario [pk]	nome_funcionario, cargo_funcionario
Cliente	Id_cliente [pk]	cpf_cliente, nome_cliente, data_nascimento_cliente, email_cliente
Jogo	Id_jogo [pk]	titulo_jogo, ano_lancamento_jogo, valor_diaria_jogo, genero_jogo, estoque_jogo_fisico, codigo_licenca_jogo_digital, Id_plataforma [fk], cnpj_fornecedor [fk]
Plataforma	Id_plataforma [pk]	nome_plataforma, fabricante_plataforma
Fornecedor	cnpj_fornecedor [pk]	contato_fornecedor, nome_fornecedor

## 2FN

Tabela	Chave Primária	Atributos
Aluguel	Id_aluguel [pk]	data_devolucao, tempo_aluguel, valor_total, multa, Id_funcionario [fk], Id_cliente [fk], Id_jogo [fk]
Funcionario	Id_funcionario [pk]	nome_funcionario, cargo_funcionario
Cliente	Id_cliente [pk]	cpf_cliente, nome_cliente, data_nascimento_cliente, email_cliente
Jogo	Id_jogo [pk]	titulo_jogo, ano_lancamento_jogo, valor_diaria_jogo, genero_jogo, Id_plataforma [fk]
JogoFisico	Id_jogo [pk/fk]	estoque_jogo_fisico, cnpj_fornecedor [fk]
JogoDigital	Id_jogo [pk/fk]	codigo_licenca_jogo_digital
Plataforma	Id_plataforma [pk]	nome_plataforma, fabricante_plataforma
Fornecedor	cnpj_fornecedor [pk]	contato_fornecedor, nome_fornecedor

## 3FN

<b>Tabela</b>	<b>Chave Primária</b>	<b>Atributos</b>
Aluguel	Id_aluguel [pk]	data_devolucao, tempo_aluguel, valor_total, multa, Id_funcionario [fk], Id_cliente [fk], Id_jogo [fk]
Funcionario	Id_funcionario [pk]	nome_funcionario, cargo_funcionario
Cliente	Id_cliente [pk]	cpf_cliente, nome_cliente, data_nascimento_cliente, email_cliente
Jogo	Id_jogo [pk]	titulo_jogo, ano_lancamento_jogo, valor_diaria_jogo, genero_jogo, Id_plataforma [fk]
JogoFisico	Id_jogo [pk/fk]	estoque_jogo_fisico, cnpj_fornecedor [fk]
JogoDigital	Id_jogo [pk/fk]	codigo_licenca_jogo_digital
Plataforma	Id_plataforma [pk]	nome_plataforma, fabricante_plataforma
Fornecedor	cnpj_fornecedor [pk]	contato_fornecedor, nome_fornecedor

### 3 DICIONÁRIO DE DADOS

#### 3.1 Tabela Funcionário

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Funcionário	id_funcionario	SERIAL	N/A	NOT NULL	PK	Identificador único de funcionário.
Funcionário	nome	VARCHAR	100	NOT NULL	-	Nome completo do dicionário.
Funcionário	cargo	VARCHAR	50	NOT NULL	-	Cargo exercido pelo funcionário.

#### 3.2 Tabela Cliente

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Cliente	id_cliente	SERIAL	N/A	NOT NULL	PK	Identificador único de cliente.
Cliente	cpf	VARCHAR	14	UNIQUE, NOT NULL	-	CPF do cliente (com formatação).
Cliente	nome	VARCHAR	100	NOT NULL	-	Nome completo do cliente.
Cliente	data_nascimento	DATE	N/A	NOT NULL	-	Data de nascimento para verificação de idade.
Cliente	email	VARCHAR	100	UNIQUE, NOT NULL	-	E-mail para contato.

### 3.3 Plataforma

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Plataforma	id_plataforma	SERIAL	N/A	NOT NULL	PK	Identificador único de plataforma.
Plataforma	nome	VARCHAR	100	NOT NULL	-	Nome comercial da plataforma (Playstation 5).
Plataforma	fabricante	VARCHAR	50	Permite Nulo	-	Empresa fabricante do console

						(ex: Sony, Nintendo).
--	--	--	--	--	--	-----------------------

### 3.4 Jogo

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Jogo	id_jogo	SERIAL	N/A	NOT NULL	PK	Identificador único de Jogo.
Jogo	titulo	VARCHAR	200	NOT NULL	-	Nome comercial do jogo.
Jogo	ano_lançamento	INTEGER	N/A	Permite Nulo	-	Ano em que o jogo foi lançado no mercado.
Jogo	valor_diaria	NUMERIC	10, 2	NOT NULL	-	Valor cobrado por dia de aluguel.
Jogo	genero	VARCHAR	50	Permite Nulo	-	Gênero do jogo (ex: Ação, RPG, Esporte).
Jogo	id_plataforma	INTEGER	N/A	NOT NULL	FK	Referência à tabela Plataforma.

### 3.5 Jogo Físico

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
JogoFísico	id_jogo	INTEGER	N/A	NOT NULL	PK, FK	Referência a tabela Jogo.
JogoFísico	estoque	INTEGER	N/A	NOT	-	Quantidade

				NULL, CHECK ( $\geq 0$ )		e de cópias físicas disponíveis.
--	--	--	--	--------------------------------	--	---

### 3.6 Jogo Digital

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
JogoDigital	id_jogo	INTEGER	N/A	NOT NULL	PK, FK	Referência a tabela Jogo.
JogoDigital	codigo_licenca	INTEGER	50	UNIQUE, NOT NULL	-	Código ou chave de ativação do jogo.

### 3.7 Fornecedor

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Fornecedor	cnpj	VARCHAR	18	NOT NULL	PK	CNPJ do fornecedor.
Fornecedor	nome	VARCHAR	100	NOT NULL	-	Nome fantasia ou razão.
Fornecedor	contato	VARCHAR	100	Permite Nulo	-	Informação de contato.

### 3.8 Aluguel

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Aluguel	id_aluguel	SERIAL	N/A	NOT NULL	PK	Identificador único da

						locação.
Aluguel	data_aluguel	TIMESTAMP	N/A	NOT NULL	-	Data e hora exata da locação.
Aluguel	data_devolucao	TIMESTAMP	N/A	Permite Nulo	-	Data real da entrega.
Aluguel	tempo_aluguel	INTEGER	N/A	Permite Nulo	-	Quantidades de dias combinados para o aluguel.
Aluguel	valor_total	NUMERIC	10, 2	Permite Nulo	-	Valor final a ser pago.
Aluguel	multa	NUMERIC	10, 2	Permite Nulo	-	Valor de multa aplicado em caso de atraso.
Aluguel	id_funcionario	INTEGER	N/A	NOT NULL	FK	Funcionário que realizou o atendimento.
Aluguel	id_cliente	INTEGER	N/A	NOT NULL	FK	Cliente que realizou a locação.

### 3.9 Tabela Abastece

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Abastece	cnpj_funcionario	VARCHAR	18	NOT NULL	PK, FK	Referência ao Fornecedor.
Abastece	id_jogo_fisico	INTEGER	N/A	Permite Nulo	PK, FK	Referência ao Jogo Físico.

Abastece	data_abastecimento	DATE	N/A	NOT NULL	-	Data em que os jogos foram recebidos.
Abastece	quantidade	INTEGER	N/A	NOT NULL, CHECK (> 0)	-	Quantidades de unidades recebidas no abastecimento.

### 3.10 Tabela Inclui

Nome da tabela	Nome da coluna	Tipo de dados	Tamanho	Restrições	Chaves	Descrição
Inclui	id_aluguel	INTEGER	N/A	NOT NULL	PK, FK	Vincula o item ao aluguel pai.
Inclui	id_jogo	INTEGER	N/A	NOT NULL	PK, FK	Vincula o item ao jogo alugado.
Inclui	valor_unitario	NUMERIC	10, 2	NOT NULL	-	Preço do jogo unitário no momento do aluguel.
Inclui	quantidade	INTEGER	N/A	NOT NULL, DEFAULT 1, CHECK (>0)	-	Quantidade de cópias do mesmo jogo alugado.



Este link <https://github.com/gabinaiare/Projeto-Banco-de-Dados> dá acesso ao repositório do github onde está todas os pontos citados há abaixo

## **4 IMPLEMENTAÇÃO E RECURSOS AVANÇADOS**

### **4.1 OPERAÇÕES BÁSICOS SQL**

### **4.2 VIEWS**

### **4.3 GATILHOS (TRIGGERS)**

### **4.4 FUNÇÕES**

## **5 PLANO DE TESTES**

### **5.1 TESTES DE INSERÇÃO**

### **5.2 TESTES DE REMOÇÃO**

### **5.3 TESTES DE LISTAGEM (SELECT)**

### **5.4 TESTES DE ALTERAÇÃO (UPDATE)**

### **5.5 TESTES DE VIEWS**

### **5.6 TESTES DE GATILHOS (TRIGGERS)**

### **5.7 TESTE DE FUNÇÕES**

## **6 CONCLUSÃO**

Neste trabalho, construímos um banco de dados completo para uma locadora de jogos, contemplando todas as funcionalidades essenciais cadastro de clientes e funcionários, registro de jogos físicos, controle de estoque, registro de alugueis e devoluções, cálculo de multas por atraso, e geração de relatórios por meio de views. A modelagem incluiu chaves primárias, restrições de integridade, triggers para automação de estoque e manutenção de consistência, e

funções que encapsulam a lógica de negócio. Com isso, alcançamos um sistema robusto, consistente e pronto para suportar operações reais de locação.

Durante o desenvolvimento, obtivemos diversos aprendizados importantes:

- A modelagem de dados exige atenção à integridade referencial, integridade de domínio (tipos, constraints) e às regras de negócio.
- O uso de triggers e funções no banco de dados permite automatizar tarefas críticas (como controle de estoque e cálculo de multa), reduzindo o risco de erros manuais e mantendo a consistência.
- A normalização do banco, ou seja, a estruturação apropriada de tabelas e relacionamentos é fundamental para evitar redundâncias e permitir expansibilidade do sistema.
- A elaboração de testes, inclusive de cenários que devem falhar, é essencial para validar que o modelo realmente protege contra inconsistências e violações de regras e não apenas “funciona quando tudo vai bem”.

Por outro lado, enfrentamos desafios relevantes:

- Garantir que os triggers e functions fossem escritos com a lógica correta e na ordem certa (função antes do trigger), para evitar erros de execução.
- Lidar com os casos de exclusão: manter integridade referencial exigiu decisões sobre uso de ON DELETE CASCADE vs. restrições rígidas.
- Equilibrar a flexibilidade do sistema com a segurança dos dados por exemplo, permitir exclusões pode facilitar manutenção, mas pode também apagar dados importantes; por isso, a escolha cuidadosa das constraints foi essencial.
- Planejar e implementar testes para cenários de sucesso e de falha, o que demanda pensar em situações adversas que nem sempre surgem durante uso normal.

## 7 REFERÊNCIAS (Se houver)

- <https://www.youtube.com/watch?v=1RXT1r1zLIU&t=386s>
- <https://spaceprogrammer.com/bd/normalizando-um-banco-de-dados-por-meio-das-3-principais-formas/>