

**SENAI GASPAR RICARDO JÚNIOR**  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**Gizela alves Pieri Santos**

**Graciele Garglioni**

**Isabelle Nastri Sales**

**Isadora da Silva Zanardo**

**ANDRÉ CASSULINO ARAÚJO SOUZA**

**BANCO DE DADOS**

**Projeto Sistema de Gerenciamento de Biblioteca**

**SOROCABA – SP**

**2025**

## Sumário

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>3</b>
<b>2</b>	<b>MODELAGEM CONCEITUAL.....</b>	<b>4</b>
<b>3</b>	<b>MODELAGEM LÓGICA .....</b>	<b>5</b>
<b>4</b>	<b>ESTRUTURA DO BANCO DE DADOS .....</b>	<b>6</b>
<b>5</b>	<b>MANIPULAÇÃO DE DADOS .....</b>	<b>8</b>
<b>6</b>	<b>CONTROLE DE ACESSO.....</b>	<b>10</b>
<b>7</b>	<b>CONTROLE DE TRANSAÇÕES.....</b>	<b>11</b>
<b>8</b>	<b>CONCLUSÃO.....</b>	<b>14</b>
<b>9</b>	<b>REFERÊNCIAS .....</b>	<b>15</b>

## **1 INTRODUÇÃO**

O objetivo deste projeto é aplicar de forma prática os conhecimentos adquiridos ao longo da disciplina de Banco de Dados, por meio da concepção, modelagem e implementação de um sistema de banco de dados voltado para a gestão de uma biblioteca. A escolha do nosso projeto foi baseada a fim de possibilitar uma exploração abrangente dos principais conceitos relacionados ao armazenamento, organização e recuperação de dados, considerando aspectos como o cadastro de livros na faculdade, controle de empréstimos, registro de alunos e gerenciamento de acervos.

Durante o desenvolvimento do sistema, foram seguidas as etapas de modelagem conceitual, representadas por um Diagrama Entidade-Relacionamento (DER), além da normalização das tabelas até a Terceira Forma Normal (3FN), com o objetivo de garantir a consistência e eliminar redundâncias nos dados. Adicionalmente, foram elaborados scripts SQL utilizando as linguagens DDL (Data Definition Language), DML (Data Manipulation Language) e DQL (Data Query Language), fundamentais para a criação da estrutura do banco, bem como para a manipulação e consulta das informações.

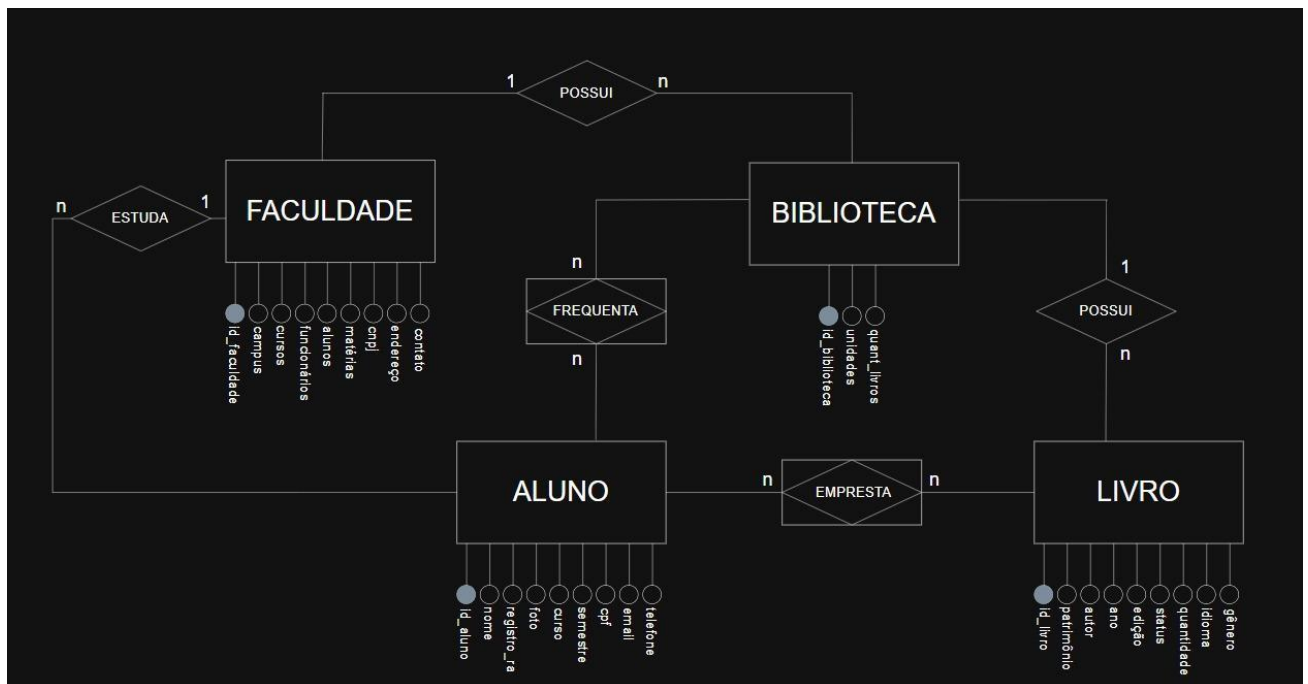
Para aprofundar o domínio das linguagens de banco de dados e atender aos critérios de bonificação, foram incluídas funcionalidades que utilizam comandos de DCL (Data Control Language) para o gerenciamento de permissões, além de DTL (Data Transaction Language) para assegurar a integridade das transações realizadas.

O projeto foi desenvolvido em conformidade com as normas da ABNT e representa uma oportunidade de consolidar os conteúdos estudados e concedidos em sala de aula por meio de atividades na prática, que nos ajudou a formar uma base sólida, para realizar este trabalho.

## 2 MODELAGEM CONCEITUAL

### DER:

O modelo DER representa um sistema acadêmico criado para gerenciar alunos, faculdades, bibliotecas e empréstimos de livros. Ele cobre a relação dos alunos com a instituição de ensino, indicando que eles estudam nela, bem como seu acesso às bibliotecas, que frequentam. Além disso, o modelo inclui o controle dos empréstimos de livros. A estrutura organizacional é composta pela faculdade, biblioteca e acervo.



### 3 MODELAGEM LÓGICA

O modelo do banco de dados da biblioteca foi projetado seguindo a Terceira Forma Normal (3FN), respeitando os princípios de normalização para evitar redundâncias e assegurar a integridade dos dados. Na Primeira Forma Normal (1FN), os dados são organizados em valores atômicos, eliminando repetições. Na Segunda Forma Normal (2FN), todas as dependências não-chave são associadas à chave primária, utilizando tabelas intermediárias para gerenciar relacionamentos muitos-para-muitos. A 3FN é alcançada ao remover dependências transitivas, garantindo que todos os atributos dependam exclusivamente da chave primária. Dessa maneira, o banco de dados é estruturado de forma eficaz, livre de redundâncias e capaz de manter a integridade das informações.

#### Normalização do Modelo Entidade Relacionamento

A normalização do modelo entidade relacionamento da biblioteca foi conduzida conforme as três primeiras formas normais (1FN, 2FN e 3FN), visando garantir a integridade dos dados e eliminar redundâncias.

1FN: Todos os atributos são atômicos, incluindo os campos “telefone,” “email,” “cpf” e “campus” do cliente.

2FN: Os atributos não chave dependem integralmente da chave primária. Para gerenciar os relacionamentos muitos-para-muitos entre CLIENTE, LIVRO e BIBLIOTECA, foram estabelecidas tabelas intermediárias (CLIENTE\_BIBLIOTECA e EMPRESTIMO) que utilizam chaves compostas.

3FN: As dependências transitivas foram eliminadas. Por exemplo, o atributo “quant\_livros” na tabela BIBLIOTECA depende exclusivamente da chave primária.

Assim, a normalização foi aplicada de maneira eficaz, resultando em um banco de dados eficiente que reduz redundâncias e melhora a integridade dos dados. O modelo também reflete as melhores práticas de modelagem relacional.

#### 4 ESTRUTURA DO BANCO DE DADOS

--create

```
create table bibilioteca(  
id serial primary key,  
nome_campus varchar(50),  
data_emprestimo date,  
data_devolucao date  
);
```

--create

```
create table cliente(  
cliente_id serial primary key,  
foto varchar(20),  
cpf varchar(11),  
nome varchar(100) not null,  
email varchar(100) unique,  
telefone varchar(20),  
curso varchar(20)  
);
```

--create

```
create table livro(  
codigo_id serial primary key,  
autor varchar(100),  
ano_data date,  
edicao varchar(50),  
quantidade_livro numeric(10,1) not null,  
status_do_aluguel varchar(100),  
idioma varchar (50),  
Gênero varchar(20)  
);
```

-- create: tabela que relaciona clientes, livros e registros de empréstimo

```
create table empréstimo (  
    emprestimo_id serial primary key,  
    cliente_id int references cliente(cliente_id),  
    codigo_id int references livro(codigo_id),
```

```
biblioteca_id int references bibilioteca(id)
);
```

## 5 MANIPULAÇÃO DE DADOS

### DML:

-- Inserir clientes

```
INSERT INTO aluno (telefone, email, cpf, semestre, curso, foto, registro_ra, nome, id_aluno)
```

```
VALUES (('11)99999-0000', 'isazanardi@gmail.com', '11000000', 'semestre 2', 'Ads', 'fotoq1.jpg', 0000, 'Isadora', 09);
```

-- Inserir livros

```
INSERT INTO livro (genero, idioma, quantidade, status, edicao, ano, autor, patrimonio, id_livro)
```

```
VALUES ('Romance', 'Português', 10, 'Disponível', '2ª', 2020, 'Machado de Assis', 'BR12345', 1);
```

-- Inserir biblioteca

```
INSERT INTO biblioteca (quant_livros, unidades, id_biblioteca)
```

```
VALUES (1500, 'Campus Centro', 1);
```

- - Inserir Faculdade

```
INSERT INTO faculdade (contato, endereco, cnpj, materias, alunos, funcionarios, cursos, campus, id_faculdade)
```

```
VALUES (('11) 4002-8922', 'Rua das Flores, 123 - São Paulo, SP', '12.345.678/0001-90', 35, 1200, 80, 10, 'Centro', 1);
```

-- Inserir empréstimo

```
INSERT INTO emprestimo (cliente_id, codigo_id, biblioteca_id)
```

```
VALUES (1, 1, 1);
```

### DQL

Listar todos os clientes cadastrados:

```
SELECT * FROM cliente;
```

Listar os livros disponíveis para empréstimo:



SELECT \* FROM livro

WHERE status\_do\_aluguel = 'Disponível';

Mostrar os dados de todos os empréstimos:

SELECT

e.emprestimo\_id,

c.nome AS nome\_cliente,

l.autor AS autor\_livro,

b.nome\_campus AS biblioteca

FROM

emprestimo e

JOIN cliente c ON e.cliente\_id = c.cliente\_id

JOIN livro l ON e.codigo\_id = l.codigo\_id

JOIN biblioteca b ON e.biblioteca\_id = [b.id](#);

Quantidade total de livros cadastrados:

SELECT SUM(quantidade\_livro) AS total\_livros FROM livro;

Ver todos os clientes que já realizaram algum empréstimo:

SELECT DISTINCT c.nome

FROM cliente c

JOIN emprestimo e ON c.cliente\_id = e.cliente\_id;

Mostrar a quantidade de livros emprestados por cliente:

SELECT c.nome, COUNT(e.emprestimo\_id) AS livros\_emprestados

FROM cliente c

JOIN emprestimo e ON c.cliente\_id = e.cliente\_id

GROUP BY c.nome;

## 6      CONTROLE DE ACESSO

### **DCL:**

-- Criação de Usuários

CREATE USER bibliotecario IDENTIFIED BY 'senha123';

CREATE USER leitor IDENTIFIED BY 'senha456';

-- Atribuição de Permissões

GRANT SELECT, INSERT, UPDATE, DELETE ON livros TO bibliotecario;

GRANT SELECT ON livros TO leitor;

-- Revogação de Permissões

REVOKE DELETE ON livros FROM bibliotecario;

-- Verificação de Permissões

SHOW GRANTS FOR bibliotecario;

SHOW GRANTS FOR leitor;

## 7 CONTROLE DE TRANSAÇÕES

### DTL:

Função para registrar empréstimo

```
CREATE OR REPLACE FUNCTION registrar_emprestimo(
    p_livro_id INT,
    p_membro_id INT,
    p_data_emprestimo DATE,
    p_data_devolucao_prevista DATE
)
RETURNS TEXT AS $$
DECLARE
    v_quantidade_disponivel INT;
BEGIN
    -- Verifica a quantidade disponível
    SELECT quantidade_disponivel INTO v_quantidade_disponivel
    FROM livros
    WHERE id_livro = p_livro_id;
    IF v_quantidade_disponivel > 0 THEN
        -- Atualiza quantidade
        UPDATE livros
        SET quantidade_disponivel = quantidade_disponivel - 1
        WHERE id_livro = p_livro_id;
        -- Insere empréstimo
        INSERT INTO emprestimos (id_livro, id_membro, data_emprestimo,
        data_devolucao_prevista, status)
        VALUES (p_livro_id, p_membro_id, p_data_emprestimo,
        p_data_devolucao_prevista, 'EMPRESTADO');
        RETURN 'Empréstimo registrado com sucesso!';
    ELSE
        RETURN 'Não há livros disponíveis.';
```

```

    END IF;
END;

$$ LANGUAGE plpgsql;

-- Exemplo de chamada correta (fora da função):
-- BEGIN;
-- SELECT registrar_emprestimo(1, 1, '2025-05-30', '2025-06-04');
-- COMMIT;
-- Função para registrar devolução
CREATE OR REPLACE FUNCTION registrar_devolucao(
    p_emprestimo_id INT,
    p_data_devolucao DATE
)
RETURNS TEXT AS $$
DECLARE
    v_livro_devolvido_id INT;
    v_rows_updated INT;
BEGIN
    -- Atualiza o empréstimo se ainda não foi devolvido
    UPDATE empréstimos
    SET
        status = 'DEVOLVIDO',
        data_devolucao = p_data_devolucao
    WHERE
        id_emprestimo = p_emprestimo_id
        AND status = 'EMPRESTADO'
    RETURNING id_livro INTO v_livro_devolvido_id;
    IF FOUND THEN
        -- Atualiza a quantidade de livros disponíveis
        UPDATE livros
        SET quantidade_disponivel = quantidade_disponivel + 1

```

```
WHERE id_livro = v_livro_devolvido_id;

RETURN 'Devolução registrada com sucesso!';

ELSE

    RETURN 'Empréstimo não encontrado ou já devolvido.';

END IF;

EXCEPTION

    WHEN OTHERS THEN

        RETURN 'Ocorreu um erro ao registrar a devolução: ' || SQLERRM;

END;

$$ LANGUAGE plpgsql
```

## 8 CONCLUSÃO

Desenvolver o projeto de banco de dados proporcionou uma aplicação prática e aprofundada dos principais conceitos da área, abrangendo desde a modelagem conceitual até a implementação e manipulação de dados em um sistema de gerenciamento para uma biblioteca acadêmica. A elaboração do Diagrama Entidade-Relacionamento (DER) permitiu definir claramente entidades, atributos e relacionamentos, servindo como uma base sólida para a modelagem lógica e física do banco de dados.

O modelo foi normalizado até a Terceira Forma Normal (3FN), o que garantiu a integridade referencial e eliminou redundâncias, resultando em um esquema relacional eficiente e de fácil manutenção. As estruturas de dados foram implementadas com comandos DDL, possibilitando a criação de tabelas com chaves primárias e estrangeiras adequadas às necessidades do sistema.

Foram realizadas operações DML para inserção, atualização e remoção de dados, além de consultas DQL para extrair informações relevantes, assegurando funcionalidades como controle de empréstimos, cadastro de clientes e gerenciamento do acervo de livros. Também foram explorados comandos DCL para gerenciar permissões de usuários, reforçando a segurança e o controle de acesso, enquanto as transações foram geridas com comandos DTL para garantir a confiabilidade das operações críticas, como registro de empréstimos e devoluções.

Em conclusão, o projeto alcançou seus objetivos pedagógicos e técnicos, consolidando o conhecimento em banco de dados relacional através de uma abordagem prática e aplicada. A estrutura resultante é escalável, segura e alinhada às boas práticas de desenvolvimento, estando preparada para futuras expansões ou integrações com sistemas de front-end e aplicações web.

## 9 REFERÊNCIAS

- <https://github.com/profAndreSouza/Material/tree/main/Banco%20de%20Dados>
- <https://www.jdevtreinamento.com.br/ddl-dml-linguagens-de-banco-de-dados/>
- <https://blog.grancursosonline.com.br/comandos-dtl-em-banco-de-dados-o-que-voce-precisa-saber-para-concursos/>
- [https://www.alura.com.br/artigos/normalizacao-banco-de-dados-estrutura?utm\\_term=&utm\\_campaign=topo-aon-search-gg-dsa-artigos\\_conteudos&utm\\_source=google&utm\\_medium=cpc&campaign\\_id=11384329873\\_164240702375\\_703853654617&utm\\_id=11384329873\\_164240702375\\_703853654617&hsa\\_acc=7964138385&hsa\\_cam=topo-aon-search-gg-dsa-artigos\\_conteudos&hsa\\_grp=164240702375&hsa\\_ad=703853654617&hsa\\_src=g&hsa\\_tgt=dsa-2276348409543&hsa\\_kw=&hsa\\_mt=&hsa\\_net=google&hsa\\_ver=3&gad\\_source=1&gad\\_campaignid=11384329873&gbraid=0AAAAADpqZIC2rAa3Am1hTEz9QedC4BuL4&gclid=Cj0KCQjwu7TCBhCYARIsAM\\_S3Ng2eSjoi-dzfSu-z2cdto5jh2um\\_0plfOmZCHAK5ymN2wIJTd46kCYaAhEIEALw](https://www.alura.com.br/artigos/normalizacao-banco-de-dados-estrutura?utm_term=&utm_campaign=topo-aon-search-gg-dsa-artigos_conteudos&utm_source=google&utm_medium=cpc&campaign_id=11384329873_164240702375_703853654617&utm_id=11384329873_164240702375_703853654617&hsa_acc=7964138385&hsa_cam=topo-aon-search-gg-dsa-artigos_conteudos&hsa_grp=164240702375&hsa_ad=703853654617&hsa_src=g&hsa_tgt=dsa-2276348409543&hsa_kw=&hsa_mt=&hsa_net=google&hsa_ver=3&gad_source=1&gad_campaignid=11384329873&gbraid=0AAAAADpqZIC2rAa3Am1hTEz9QedC4BuL4&gclid=Cj0KCQjwu7TCBhCYARIsAM_S3Ng2eSjoi-dzfSu-z2cdto5jh2um_0plfOmZCHAK5ymN2wIJTd46kCYaAhEIEALw)

