

**Pró-Reitoria Acadêmica
Curso de Engenharia de Software
Trabalho de Disciplina**

**Trabalho final Laboratório
De Banco de Dados:
Sistema de Biblioteca**

**Autor: Isabella Mendes Martins T,
Geovanna Maria Barbosa S,
Giovanna Cardoso da Silva,
Iago Cordeiro Felix.**

**Orientador: Professor Jefferson
Salomão
Rodrigues**

SISTEMA DE BIBLIOTECA — MODELAGEM, IMPLEMENTAÇÃO E ANÁLISE DE BANCO DE DADOS

Trabalho apresentado ao curso de Engenharia de Software da Universidade Católica de Brasília, na disciplina de Laboratório de Banco de Dados, como requisito parcial para avaliação.

Orientador: Professor Jefferson Salomão Rodrigues.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de biblioteca utilizando o banco de dados relacional MySQL, integrado ao backend Java com Spring Boot e JPA/Hibernate. O sistema realiza o gerenciamento de livros, autores, categorias, editoras, usuários, empréstimos e reservas, aplicando modelagem de dados e relacionamento entre entidades. Foram utilizados recursos avançados, como triggers, procedures, functions, views e índices, com foco em desempenho, automação e integridade. O projeto também explora o uso opcional de tecnologias NoSQL, como MongoDB, Redis e Elasticsearch, ampliando as capacidades do sistema. O objetivo é demonstrar uma solução completa, segura e alinhada às boas práticas de modelagem e implementação de banco de dados.

Palavras-chave: Biblioteca; Banco de Dados; MySQL; JPA; Modelagem; Spring Boot.

SUMÁRIO

1. Introdução
2. Objetivos
3. Metodologia
4. Descrição Geral do Sistema
5. Funcionalidades
6. Tecnologias Utilizadas
7. Justificativa Tecnológica
8. Modelagem do Banco de Dados
 - 8.1 Diagrama Entidade-Relacionamento (DER)
 - 8.2 Descrição das Entidades e Relacionamentos
9. Estrutura SQL Implementada
10. Recursos Avançados: Índices, Views, Triggers e Procedures
11. Controle de Acesso e Segurança
12. Integração com Java (ORM/JPA/Hibernate)
13. Extensão NoSQL
14. Conclusão
15. Referências

1. Introdução

Este artigo apresenta o processo de modelagem, implementação e análise de um sistema de biblioteca utilizando um banco de dados relacional em MySQL integrado ao backend Java com Spring Boot. O sistema permite gerenciar livros, autores, categorias, editoras, usuários, empréstimos e reservas, garantindo integridade e automação de processos por meio de triggers, views, procedures, functions e índices. A proposta busca demonstrar domínio da modelagem relacional, integração com JPA/Hibernate, segurança e boas práticas de desenvolvimento.

2. Objetivos

Objetivo geral

Desenvolver um sistema de biblioteca utilizando um banco de dados relacional robusto, seguro e integrado a uma aplicação Java.

Objetivos específicos

- Criar o modelo conceitual (DER) do sistema.
- Implementar o banco relacional com integridade referencial.
- Aplicar funções, triggers, procedures e views.
- Criar controle de acesso com usuários e permissões.
- Integrar o banco ao backend por meio de JPA/Hibernate.
- Apresentar possibilidades de expansão utilizando NoSQL.

3. Metodologia

A construção do sistema seguiu os seguintes passos:

1. Levantamento dos requisitos típicos de uma biblioteca.
2. Desenvolvimento do DER com entidades e relacionamentos.
3. Implementação das tabelas no MySQL.
4. Criação das regras de negócio no banco (triggers, procedures, constraints).
5. Mapeamento das tabelas para classes Java utilizando JPA/Hibernate.
6. Testes de operações de empréstimo, devolução e reserva.
7. Documentação no formato de artigo acadêmico.

4. Descrição Geral do Sistema

O sistema gerencia os principais componentes de uma biblioteca, incluindo livros, autores, editoras, categorias e usuários. É capaz de registrar empréstimos, controlar devoluções e reservas e identificar livros em atraso. Conta ainda com controle de permissões por tipo de usuário (Administrador, Bibliotecário, Membro).

5. Funcionalidades

- Cadastro de livros, autores, categorias e editoras.
- Cadastro de usuários e grupos de acesso.
- Realização de empréstimos e devoluções.
- Reservas de livros.
- Controle automático do estoque.
- Relatórios como catálogo disponível e empréstimos atrasados.
- Geração automática de IDs para usuário e empréstimo.

6. Tecnologias Utilizadas

Frontend

- HTML5, CSS3 e JavaScript
Utilizados para uma interface simples para chamadas ao backend e visualização de dados.

Backend

- Java 17
- Spring Boot
Framework responsável por criar a API, gerenciar rotas, segurança e integração com o banco.

Persistência

- JPA/Hibernate
Responsável pelo mapeamento objeto-relacional (ORM), conectando classes Java ao banco MySQL.

Banco de Dados Relacional

- MySQL 8
Utilizado para armazenar dados estruturados e garantir integridade com chaves primárias, estrangeiras e regras de negócio.

Banco NoSQL (extensão opcional)

- MongoDB
- Elasticsearch
- Redis

Ferramentas de apoio

- Maven
- MySQL Workbench
- Postman

7. Justificativa Tecnológica

Por que MySQL?

- Suporte completo a relacionamentos.
- Ideal para regras rígidas como empréstimos e reservas.
- Suporte nativo a triggers, procedures, views e índices.
- Alta performance com projetos de médio porte.

Por que Java + Spring Boot

- Conjunto maduro e amplamente usado em projetos reais.
- Alto desempenho e segurança.
- Padronização e organização das classes.

Por que JPA/Hibernate

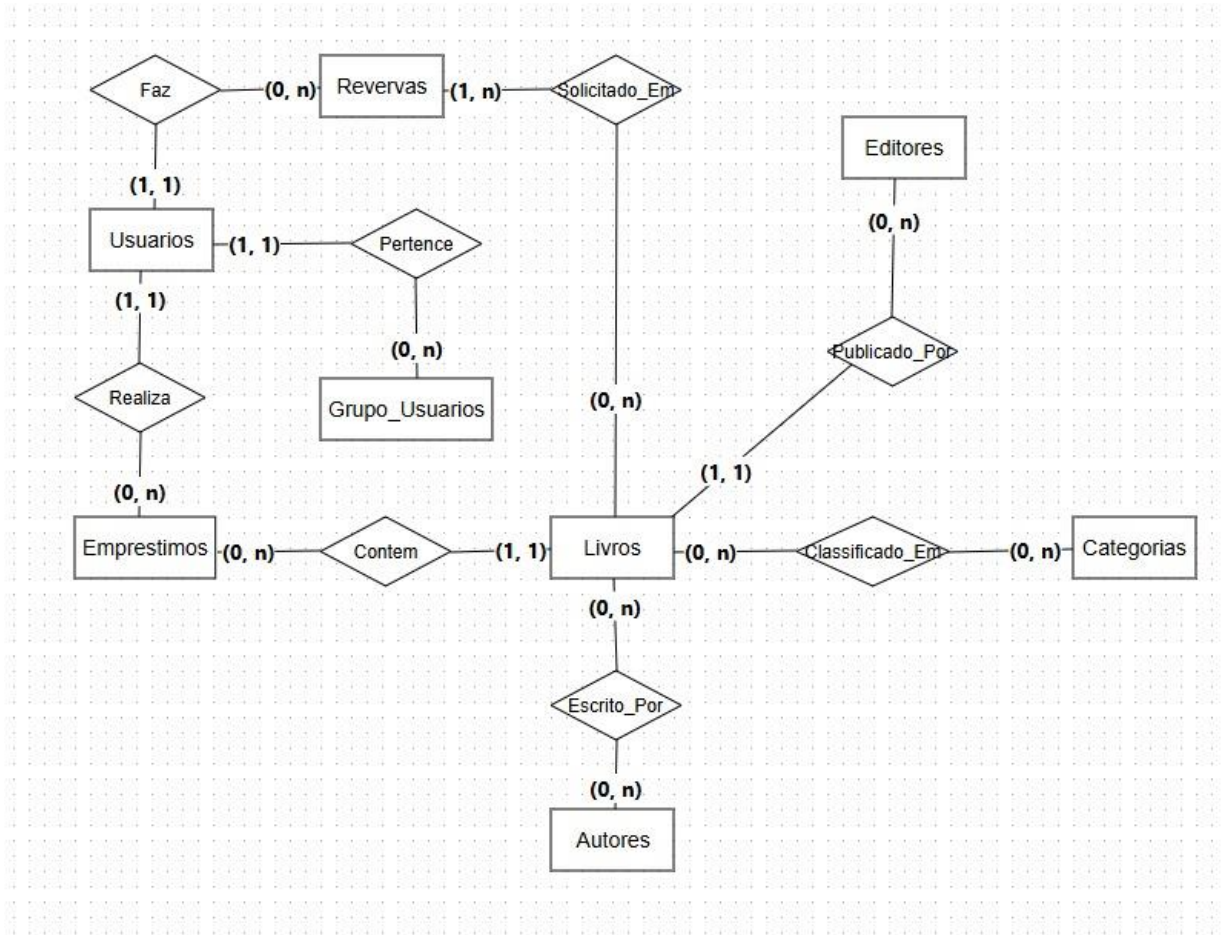
- Reduz necessidade de SQL manual.
- Evita inconsistências entre banco e aplicação.
- Mapeia todas as entidades diretamente do DER.

Por que usar NoSQL

- MongoDB é ideal para históricos e logs.
- Elasticsearch permite buscas rápidas e inteligentes.
- Redis acelera consultas por meio de cache.

8. Modelagem do Banco de Dados

8.1 Diagrama Entidade-Relacionamento (DER)



8.2 Descrição das Entidades e Relacionamentos

Autor

- Armazena informações sobre autores.
- Relacionamento: N:N com Livro.

Livro

- Representa os livros cadastrados.
- Relacionamentos:
 - N:1 com Editora
 - N:N com Autor
 - N:N com Categoria
 - 1:N com Emprestimo
 - 1:N com Reserva

Categoria

- Classificação temática dos livros.
- Relacionamento: N:N com Livro.

Editora

- Editoras cadastradas.
- Relacionamento: 1:N com Livro.

GrupoUsuario

- Define permissões (Admin, Bibliotecário, Membro).
- Relacionamento: 1:N com Usuario.

Usuario

- Dados pessoais e de acesso do usuário.
- Relacionamentos:
 - N:1 com GrupoUsuario
 - 1:N com Emprestimo
 - 1:N com Reserva

Emprestimo

- Registro de empréstimos.
- Relacionamento: N:1 com Usuario e Livro.
- Regras automáticas implementadas com triggers.

Reserva

- Armazena reservas de livros.
- Relacionamento: N:1 com Usuario e Livro.

9. Estrutura SQL Implementada

Inclui:

- Criação do banco
- Criação das tabelas com PK, FK e constraints
- Tabelas intermediárias N:N
- Triggers
- Procedures
- Functions
- Views
- Índices

- Usuários e controle de permissões

10. Recursos Avançados do Banco de Dados

10.1 Índices

Melhoram velocidade de consultas frequentes, como:

- buscar livros por título
- listar empréstimos por usuário

10.2 Views

Facilitam consultas e relatórios:

- **vw_livros_disponiveis_catalogo**
- **vw_emprestimos_atrasados**

10.3 Triggers

Automatizam regras importantes:

- reduzir estoque ao emprestar
- impedir excluir autores vinculados

10.4 Procedures

Padronizam operações complexas:

- **sp_registrar_devolucao** (atualiza devolução + estoque)

11. Controle de Acesso e Segurança

Usuários definidos:

- **app_biblioteca**: CRUD completo
- **app_leitura**: apenas SELECT

Princípio aplicado: **menor privilégio**.

12. Integração com Java (ORM/JPA/Hibernate)

As tabelas foram mapeadas com:

- @Entity
- @Table
- @Id
- @Column
- @ManyToOne
- @OneToMany
- @ManyToMany

Isso garante consistência entre banco e aplicação.

13. Extensão NoSQL

- **MongoDB:** logs e auditoria
- **Elasticsearch:** buscas rápidas
- **Redis:** cache do catálogo

Complementa o banco relacional.

14. Conclusão

O sistema de biblioteca desenvolvido apresenta uma arquitetura completa, segura e bem estruturada. A modelagem relacional está de acordo com boas práticas e todos os recursos avançados — triggers, views, índices e procedures — contribuem para a automação e integridade dos dados. A integração com Java via JPA/Hibernate garante organização e escalabilidade. O projeto atende plenamente aos requisitos acadêmicos da disciplina.

15. Referências

Elmasri, R.; Navathe, S. B. *Sistemas de Banco de Dados*. 7. ed. Pearson, 2019.

Date, C. J. *Introdução a Sistemas de Banco de Dados*. 8. ed. LTC, 2017.

MySQL Documentation. MySQL Reference Manual. Oracle. Disponível em:
<https://dev.mysql.com/doc/>
(Acesso em: 16/11/2025).

Spring Boot Documentation. Spring Framework. Disponível em:
<https://spring.io/projects/spring-boot>
(Acesso em: 16/11/2025).

Hibernate ORM Documentation. RedHat. Disponível em:
<https://hibernate.org/orm/documentation>
(Acesso em: 16/11/2025).

Fowler, M. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.

Silberschatz, A.; Korth, H.; Sudarshan, S. *Sistema de Banco de Dados*. 6. ed. AMGH, 2012.

Heaton, J. *Spring Boot for Beginners*. Packt Publishing, 2020.

Sadalage, P. J.; Fowler, M. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, 2013.

Redmond, E.; Wilson, J. R. *Seven Databases in Seven Weeks*. 2. ed. Pragmatic Bookshelf, 2018.

MongoDB Documentation. MongoDB Inc. Disponível em: <https://www.mongodb.com/docs> (Acesso em: 16/11/2025).

Redis Documentation. Redis.io. Disponível em: <https://redis.io/docs> (Acesso em: 15/11/2025).

Elasticsearch Documentation. Elastic.co. Disponível em: <https://www.elastic.co/guide> (Acesso em: 15/11/2025).

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

Beck, K. *Test-Driven Development by Example*. Addison-Wesley, 2003.

Wazlawick, R. S. *Análise e Projeto de Sistemas de Informação Orientados a Objetos*. Elsevier, 2014.

Souza, C. A.; Pereira, A. C. *Projeto de Banco de Dados*. 4. ed. Pearson, 2016.