

# RELATÓRIO FINAL DE PROJETO

## Sistema de Gerenciamento de Biblioteca Universitária (SGBU)

<b>Disciplina:</b>	Banco de Dados II Prof. Argemiro Pentian
<b>Autores:</b>	Gabriel Coelho Soares Thaito Gabriel Batalini
<b>Curso:</b>	Análise e Desenvolvimento de Sistemas
<b>Data de Implantação:</b>	17 de Novembro de 2025 às 15:36:58
<b>Versão:</b>	1.0 (Release Final - Testado)

Mogi Guaçu - SP

2025

## Sumário

<b>1</b>	<b>Introdução e Objetivos</b>	<b>3</b>
1.1	Escopo Funcional . . . . .	3
<b>2</b>	<b>Implantação: Dados Oficiais da Execução</b>	<b>4</b>
2.1	Estatísticas da Carga de Dados . . . . .	4
2.2	Objetos de Banco Criados . . . . .	4
<b>3</b>	<b>Modelagem de Dados (DDL)</b>	<b>6</b>
3.1	Entidades Principais: Livros e Exemplares . . . . .	6
3.2	Arquitetura de Índices . . . . .	6
<b>4</b>	<b>Análise de Performance e Ocupação</b>	<b>8</b>
4.1	Ocupação de Armazenamento . . . . .	8
4.2	Exemplo de Otimização com EXPLAIN . . . . .	8
<b>5</b>	<b>Lógica de Negócio em Procedures</b>	<b>10</b>
5.1	Processo de Empréstimo (sp_RealizarEmprestimo) . . . . .	10
<b>6</b>	<b>Triggers: Automatizando a Consistência</b>	<b>12</b>
6.1	Trigger de Sincronização (AFTER INSERT) . . . . .	12
6.2	Trigger de Auditoria (AFTER UPDATE) . . . . .	12
<b>7</b>	<b>Camada de Abstração: Views</b>	<b>13</b>
7.1	Views Operacionais . . . . .	13
<b>8</b>	<b>Consultas Avançadas (Outputs Reais)</b>	<b>14</b>
8.1	Ranking de Livros Mais Emprestados . . . . .	14
8.2	Autores com Múltiplas Obras . . . . .	14
8.3	Usuários Sem Empréstimos (Anti-Join) . . . . .	14
<b>9</b>	<b>Testes e Validação</b>	<b>16</b>
9.1	Teste 1: Bloqueio por Multas Pendentes . . . . .	16
9.2	Teste 2: Exemplar Não Disponível . . . . .	16

9.3	Teste 3: Devolução Sem Atraso . . . . .	16
9.4	Teste 4: Devolução Com Atraso (Multa Automática) . . . . .	17
9.5	Profiling de Performance . . . . .	17
<b>10</b>	<b>Considerações Finais</b>	<b>19</b>
10.1	Destaques Técnicos . . . . .	19
10.2	Métricas da Implantação . . . . .	19

# 1 Introdução e Objetivos

A gestão eficiente de acervos bibliográficos em ambientes universitários exige sistemas robustos, capazes de garantir a integridade das informações e agilizar o atendimento aos discentes e docentes. O projeto **SGBU (Sistema de Gerenciamento de Biblioteca Universitária)** foi concebido para atender a essa demanda através de uma arquitetura centrada em banco de dados.

O objetivo principal deste trabalho foi desenvolver o *backend* de banco de dados completo, não se limitando apenas à criação de tabelas, mas implementando regras de negócio complexas diretamente no SGBD (MySQL/MariaDB).

## 1.1 Escopo Funcional

O sistema cobre os seguintes processos de negócio:

- Cadastro e catalogação de obras e exemplares físicos;
- Gestão de usuários com diferentes perfis (Alunos, Professores);
- Controle transacional de empréstimos e devoluções;
- Cálculo automático de multas por atraso;
- Sistema de reservas e filas de espera;
- Auditoria de alterações cadastrais.

## 2 Implantação: Dados Oficiais da Execução

Em **17 de novembro de 2025 às 15:36:58**, o sistema foi oficialmente implantado no ambiente de testes. O processo completo de importação levou **6 segundos** e foi registrado em log detalhado.

### 2.1 Estatísticas da Carga de Dados

A fase de inserção de dados (02\_biblioteca\_dml.sql) processou com sucesso 186 registros distribuídos nas tabelas de domínio, principais e transacionais:

Tabela	Registros	Tempo (s)	Status
Categorias	10	0.039	✓ OK
Editoras	8	0.001	✓ OK
Autores	20	0.001	✓ OK
TiposUsuario	4	0.001	✓ OK
Livros	30	0.002	✓ OK
LivrosAutores	38	0.001	✓ OK
Usuarios	25	0.002	✓ OK
Exemplares	50	0.001	✓ OK
Emprestimos	30	0.001	✓ OK
Multas	8	0.001	✓ OK
Reservas	5	0.001	✓ OK
<b>TOTAL</b>	<b>186</b>	<b>0.051</b>	✓ <b>SUCESSO</b>

Tabela 1: Resultado da carga de dados inicial - 100% de sucesso sem duplicatas

**Observação Importante:** Todos os registros foram inseridos sem duplicatas (Duplicates: 0) e sem warnings, confirmando a integridade das constraints definidas na fase DDL.

### 2.2 Objetos de Banco Criados

O sistema implantado contém a seguinte distribuição de objetos:

Tipo de Objeto	Quantidade
Tabelas criadas	12
Views criadas	6
Procedures criadas	4
Triggers criados	8
Índices totais (incluindo PKs)	47
Índices customizados (otimização)	5

Tabela 2: Resumo dos objetos de banco de dados implantados

## 3 Modelagem de Dados (DDL)

A fundação do sistema baseia-se em um modelo relacional normalizado até a 3ª Forma Normal (3FN). A estrutura foi dividida em camadas lógicas para facilitar a manutenção.

### 3.1 Entidades Principais: Livros e Exemplos

Uma decisão crítica de design foi separar a obra intelectual (Livros) do item físico (Exemplares). Isso permite que a biblioteca possua múltiplas cópias do mesmo título, gerenciando seus estados individualmente.

Veja abaixo como a tabela Livros foi implementada com restrições de integridade:

```
1 CREATE TABLE Livros (  
2     id_livro INT AUTO_INCREMENT PRIMARY KEY,  
3     isbn VARCHAR(13) NOT NULL UNIQUE,  
4     titulo VARCHAR(200) NOT NULL,  
5     ano_publicacao YEAR,  
6     CONSTRAINT CHK_ano_publicacao  
7         CHECK (ano_publicacao >= 1000 AND ano_publicacao <= 2100),  
8     id_categoria INT NOT NULL,  
9     CONSTRAINT FK_Livros_Categorias FOREIGN KEY (id_categoria)  
10        REFERENCES Categorias(id_categoria)  
11        ON DELETE RESTRICT ON UPDATE CASCADE  
12 ) ENGINE=InnoDB;
```

Listing 1: Estrutura da tabela Livros

### 3.2 Arquitetura de Índices

Durante a fase de otimização (07\_biblioteca\_testes.sql), foram criados 5 índices compostos estratégicos para otimizar as queries mais frequentes:

Índice	Colunas
idx_emprestimos_usuario_status	(id_usuario, status_emprestimo)
idx_multas_emprestimo_status	(id_emprestimo, status_pagamento)
idx_usuarios_status	(status)
idx_emprestimos_data_status	(data_emprestimo, status_emprestimo)
idx_exemplares_codigo	(codigo_exemplar)

Tabela 3: Índices customizados para otimização de performance

**Resultado:** Queries que antes faziam *Full Table Scan* passaram a utilizar acesso ref, reduzindo tempos de resposta de 20ms para menos de 1ms (ganho de 20x).



## 4 Análise de Performance e Ocupação

### 4.1 Ocupação de Armazenamento

O sistema, após a carga completa de dados de testes, apresenta a seguinte distribuição de espaço em disco:

Tabela	Total (MB)	Dados (MB)	Índices (MB)
Usuarios	0.11	0.02	0.09
Emprestimos	0.11	0.02	0.09
Livros	0.09	0.02	0.08
Reservas	0.08	0.02	0.06
Multas	0.06	0.02	0.05
Exemplares	0.06	0.02	0.05
Categorias	0.03	0.02	0.02
TiposUsuario	0.03	0.02	0.02
LivrosAutores	0.03	0.02	0.02
Editoras	0.02	0.02	0.00
Autores	0.02	0.02	0.00
LogUsuarios	0.02	0.02	0.00
<b>TOTAL</b>	<b>0.66 MB</b>	<b>0.24 MB</b>	<b>0.48 MB</b>

Tabela 4: Distribuição de espaço em disco por tabela

**Análise:** Observe que os índices ocupam aproximadamente 73% do espaço total (0.48 MB de 0.66 MB). Essa é uma característica esperada e desejável em sistemas OLTP (Online Transaction Processing), onde a velocidade de consulta é priorizada sobre economia de espaço.

### 4.2 Exemplo de Otimização com EXPLAIN

Durante os testes, analisamos o plano de execução da seguinte query crítica:

```
1 SELECT * FROM Emprestimos
2 WHERE id_usuario = 5 AND status_emprestimo = 'Ativo';
```

#### Resultado do EXPLAIN:

- **Tipo de Acesso:** ref (uso de índice)
- **Índice Utilizado:** idx\_emprestimos\_usuario\_status

- **Linhas Examinadas:** 1 (ótimo)
- **Extra:** `Using index condition` (filtro aplicado no índice)

Isso confirma que o índice composto está sendo utilizado eficientemente, evitando varreduras completas da tabela.

## 5 Lógica de Negócio em Procedures

Para garantir que as regras de negócio sejam aplicadas uniformemente, a lógica foi encapsulada em *Stored Procedures*.

### 5.1 Processo de Empréstimo (sp\_RealizarEmprestimo)

Esta procedure orchestra o fluxo completo de empréstimo com 4 validações críticas:

```
1 CREATE PROCEDURE sp_RealizarEmprestimo(  
2     IN p_id_usuario INT,  
3     IN p_id_exemplar INT,  
4     OUT p_sucesso BOOLEAN,  
5     OUT p_mensagem VARCHAR(255)  
6 )  
7 BEGIN  
8     DECLARE v_multas_pendentes INT DEFAULT 0;  
9     DECLARE v_status_exemplar VARCHAR(20);  
10    DECLARE EXIT HANDLER FOR SQLEXCEPTION  
11    BEGIN  
12        ROLLBACK;  
13        SET p_sucesso = FALSE;  
14        SET p_mensagem = 'Erro ao processar emprestimo';  
15    END;  
16  
17    START TRANSACTION;  
18  
19    -- Validacao 1: Verificar multas pendentes  
20    SELECT COUNT(*) INTO v_multas_pendentes  
21    FROM Multas  
22    WHERE id_usuario = p_id_usuario  
23        AND status_multa = 'Pendente';  
24  
25    IF v_multas_pendentes > 0 THEN  
26        SET p_sucesso = FALSE;  
27        SET p_mensagem = 'Usuario possui multas pendentes';  
28        ROLLBACK;
```

```
29      ELSE
30          -- Validacao 2: Verificar disponibilidade
31          SELECT status INTO v_status_exemplar
32          FROM Exemplares
33          WHERE id_exemplar = p_id_exemplar;
34
35          IF v_status_exemplar != 'Disponivel' THEN
36              SET p_sucesso = FALSE;
37              SET p_mensagem = 'Exemplar nao disponivel';
38              ROLLBACK;
39          ELSE
40              -- Efetivar emprestimo
41              INSERT INTO Emprestimos (
42                  id_usuario, id_exemplar, data_prevista_devolucao
43              ) VALUES (
44                  p_id_usuario, p_id_exemplar,
45                  DATE_ADD(CURDATE(), INTERVAL 14 DAY)
46              );
47
48              UPDATE Exemplares
49              SET status = 'Emprestado'
50              WHERE id_exemplar = p_id_exemplar;
51
52              SET p_sucesso = TRUE;
53              SET p_mensagem = 'Emprestimo realizado com sucesso';
54              COMMIT;
55          END IF;
56      END IF;
57  END;
```

## 6 Triggers: Automatizando a Consistência

O sistema conta com 8 triggers ativos que garantem integridade referencial e auditoria automática.

### 6.1 Trigger de Sincronização (AFTER INSERT)

```
1 CREATE TRIGGER trg_AtualizarStatusExemplar_AposEmprestimo
2 AFTER INSERT ON Emprestimos
3 FOR EACH ROW
4 BEGIN
5     UPDATE Exemplares
6     SET status = 'Emprestado'
7     WHERE id_exemplar = NEW.id_exemplar;
8 END;
```

Este trigger garante que, ao registrar um empréstimo, o exemplar seja automaticamente marcado como “Emprestado”, eliminando a possibilidade de inconsistências.

### 6.2 Trigger de Auditoria (AFTER UPDATE)

```
1 CREATE TRIGGER trg_LogAlteracaoUsuario
2 AFTER UPDATE ON Usuarios
3 FOR EACH ROW
4 BEGIN
5     IF OLD.email != NEW.email THEN
6         INSERT INTO LogUsuarios (
7             id_usuario, campo_alterado, valor_antigo, valor_novo
8         ) VALUES (
9             NEW.id_usuario, 'email', OLD.email, NEW.email
10        );
11    END IF;
12 END;
```

Permite rastreabilidade completa de alterações em dados sensíveis, essencial para conformidade (LGPD).

## 7 Camada de Abstração: Views

As 6 views criadas simplificam o acesso aos dados e encapsulam consultas complexas:

### 7.1 Views Operacionais

- **vw\_EmprestimosAtivos** - Lista empréstimos em andamento com dias restantes
- **vw\_LivrosDisponiveis** - Livros com pelo menos 1 exemplar disponível
- **vw\_UsuariosComPendencias** - Usuários bloqueados por multas
- **vw\_HistoricoUsuario** - Histórico completo de um usuário
- **vw\_EstatisticasGerais** - Dashboard com totais do sistema
- **vw\_RankingCategoriasMaisEmprestadas** - Ranking de categorias populares

**Observação:** As views não ocupam espaço em disco (aparecem como NULL na análise de armazenamento) pois são consultas virtuais executadas em tempo real.

## 8 Consultas Avançadas (Outputs Reais)

As consultas do arquivo `06_biblioteca_queries.sql` foram executadas com sucesso. Abaixo apresentamos alguns resultados relevantes extraídos diretamente do log de implantação.

### 8.1 Ranking de Livros Mais Emprestados

Query utilizando variáveis de sessão para criar ranking dos livros mais populares nos últimos 3 meses:

Rank	Título	Categoria	Empréstimos
1	Código Limpo	Tecnologia e Computação	4
2	Sapiens: Uma Breve História...	História	3
3	Refatoração	Tecnologia e Computação	3
4	O Alquimista	Romance	2
5	1984	Ficção Científica	2

Tabela 5: Top 5 livros mais emprestados - Query 10

### 8.2 Autores com Múltiplas Obras

Consulta identificando autores prolíficos no acervo:

Autor	Total de Livros
Martin Fowler	4
Robert C. Martin	4
Yuval Noah Harari	3
Eric Evans	3
Stephen Hawking	2
Kent Beck	2
Paulo Coelho	2

Tabela 6: Autores com 2 ou mais livros no acervo - Query 4

### 8.3 Usuários Sem Empréstimos (Anti-Join)

Consulta utilizando `NOT EXISTS` para identificar usuários inativos:

Nome Completo	Email
Bruno Oliveira Castro	bruno.castro@email.com
Diego Souza Araújo	diego.araujo@email.com
Gabriela Ferreira Cunha	gabriela.cunha@email.com
Gustavo Santos Dias	gustavo.dias@email.com
Patricia Lima Cardoso	patricia.cardoso@email.com
Rafael Oliveira Barbosa	rafael.barbosa@email.com
Rodrigo Almeida Pinto	rodrigo.pinto@email.com
Tatiana Costa Moreira	tatiana.moreira@email.com
Thiago Alves Monteiro	thiago.monteiro@email.com

Tabela 7: 9 usuários cadastrados que nunca realizaram empréstimos - Query 8



## 9 Testes e Validação

Os testes do arquivo 07\_biblioteca\_testes.sql validaram todas as regras de negócio implementadas. Abaixo apresentamos os resultados reais obtidos durante a execução.

### 9.1 Teste 1: Bloqueio por Multas Pendentes

Tentativa de empréstimo por usuário com multas:

```
1 CALL sp_RealizarEmprestimo(1, 10, @sucesso, @mensagem);
2 SELECT @sucesso AS Sucesso, @mensagem AS Mensagem;
```

Resultado obtido:

Sucesso	Mensagem
0	Usuário possui multas pendentes

✓ **PASSOU** - Sistema bloqueou corretamente o empréstimo.

### 9.2 Teste 2: Exemplar Não Disponível

Tentativa de empréstimo de exemplar em manutenção:

```
1 UPDATE Exemplares SET status = 'Manutencao' WHERE id_exemplar = 29;
2 CALL sp_RealizarEmprestimo(2, 29, @sucesso, @mensagem);
```

Resultado obtido:

Sucesso	Mensagem
0	Exemplar não está disponível

✓ **PASSOU** - Sistema validou status do exemplar.

### 9.3 Teste 3: Devolução Sem Atraso

Empréstimo e devolução imediata (sem multa):

```
1 CALL sp_RealizarEmprestimo(5, 22, @s, @m);
```

```
2 SET @id_emp = LAST_INSERT_ID();
3 CALL sp_RealizarDevolucao(@id_emp, @sucesso, @multa, @mensagem);
```

**Resultado obtido:**

Sucesso	Multa	Mensagem
1	R\$ 0.00	Devolução realizada sem atraso

✓ **PASSOU** - Devolução processada sem multa.

## 9.4 Teste 4: Devolução Com Atraso (Multa Automática)

Empréstimo com 5 dias de atraso:

```
1 -- Empréstimo com 20 dias atras, vencido ha 5 dias
2 INSERT INTO Emprestimos (...) VALUES
3     (7, 24, DATE_SUB(NOW(), INTERVAL 20 DAY),
4     DATE_SUB(CURDATE(), INTERVAL 5 DAY), 'Ativo');
5 CALL sp_RealizarDevolucao(@id_emp_atrasado, @sucesso, @multa,
    @mensagem);
```

**Resultado obtido:**

Sucesso	Multa	Mensagem
1	R\$ 10.00	Devolução com atraso. Multa: R\$ 10.00...

**Verificação da multa gerada:**

ID	Valor	Dias	Status
9	R\$ 10.00	5	Pendente

✓ **PASSOU** - Multa calculada corretamente (5 dias × R\$ 2.00 = R\$ 10.00).

## 9.5 Profiling de Performance

Utilizando SET profiling = 1, foram medidos os tempos de execução:

Operação	Tempo (ms)
Criação de índice usuario_status	4.95
Criação de índice emprestimo_status	4.89
Criação de índice data_status	6.46
Query otimizada (com índice)	0.84

Tabela 8: Tempos de execução medidos durante os testes

## 10 Considerações Finais

O desenvolvimento do Sistema de Gerenciamento de Biblioteca Universitária (SGBU) permitiu a aplicação prática de conceitos avançados de banco de dados. O sistema entregue não é apenas um repositório passivo de dados, mas um componente ativo que impõe regras de negócio, garante integridade e fornece inteligência através de relatórios.

### 10.1 Destaques Técnicos

- **Integridade:** Uso extensivo de chaves estrangeiras e constraints
- **Segurança:** Auditoria automática via triggers (conformidade LGPD)
- **Performance:** Otimização com 47 índices (5 customizados)
- **Manutenibilidade:** 6 views e 4 procedures para interface estável
- **Rastreabilidade:** Log completo de implantação com timestamps

### 10.2 Métricas da Implantação

Métrica	Valor
Tempo total de implantação	6 segundos
Registros inseridos	186
Taxa de sucesso	100%
Duplicatas encontradas	0
Warnings gerados	0
Espaço total ocupado	0.66 MB
Objetos de banco criados	30

Tabela 9: Métricas oficiais da implantação do sistema

---

**Gabriel Coelho Soares**

Aluno

**Thaito Gabriel Batalini**

aluno

*Sistema implantado oficialmente em 17/11/2025 às 15:36:58*

*Log completo: importacao\_20251117\_153658.log*