**FACULDADE SENAC MARINGÁ**

**Discente: Claudemir Dias de Oliveira**

**BANCO DE DADOS ll**

**Sistema de Gestão de Biblioteca**

**Trabalho para apresentação no curso de graduação da Faculdade Senac.**

**Docente: Joszislaine Costa**

**Maringá – Paraná**

**2025**

**DOCUMENTAÇÃO TÉCNICA**

**Sistema de Gestão de Biblioteca**

**1. INTRODUÇÃO**

**1.1 Objetivo do Sistema**

O Sistema de Gestão de Biblioteca foi desenvolvido para gerenciar o acervo de livros, cadastro de usuários, controle de empréstimos e devolução de obras, além do gerenciamento de multas por atrasos. O sistema permite um controle eficiente e automatizado das operações diárias de uma biblioteca.

**1.2 Escopo**

O sistema abrange:

* Cadastro e controle de livros e autores
* Categorização do acervo
* Gestão de usuários
* Controle de empréstimos e devoluções
* Cálculo automático de multas por atraso
* Relatórios gerenciais através de views

**2. MODELAGEM DE DADOS**

**2.1 Modelo Conceitual (MER)**

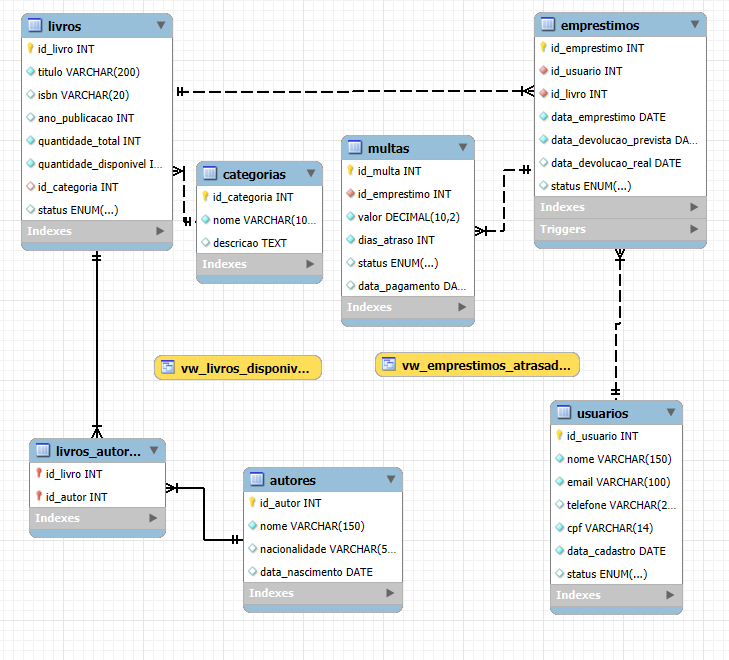
O Modelo Entidade-Relacionamento apresenta as seguintes entidades principais e seus relacionamentos:

**Entidades:**

* **USUARIOS**: Representa os usuários cadastrados na biblioteca
* **LIVROS**: Representa o acervo de livros disponíveis
* **CATEGORIAS**: Classificação dos livros por área de conhecimento
* **AUTORES**: Informações sobre os autores das obras
* **EMPRESTIMOS**: Registra as transações de empréstimo
* **MULTAS**: Armazena as penalidades por atraso

**Relacionamentos:**

* Um USUARIO pode realizar vários EMPRESTIMOS (1:N)
* Um LIVRO pode ser emprestado várias vezes (1:N)
* Uma CATEGORIA classifica vários LIVROS (1:N)
* Um AUTOR pode escrever vários LIVROS e um LIVRO pode ter vários AUTORES (N:M)
* Um EMPRESTIMO pode gerar uma MULTA (1:1)



**2.2 Modelo Lógico (DER)**

O modelo lógico implementa as seguintes tabelas com suas chaves:

**Tabela: categorias**

* id\_categoria (PK)
* nome
* descricao

**Tabela: autores**

* id\_autor (PK)
* nome
* nacionalidade
* data\_nascimento

**Tabela: livros**

* id\_livro (PK)
* titulo
* isbn
* ano\_publicacao
* quantidade\_total
* quantidade\_disponivel
* id\_categoria (FK → categorias)
* status

**Tabela: livros\_autores**

* id\_livro (PK, FK → livros)
* id\_autor (PK, FK → autores)

**Tabela: usuarios**

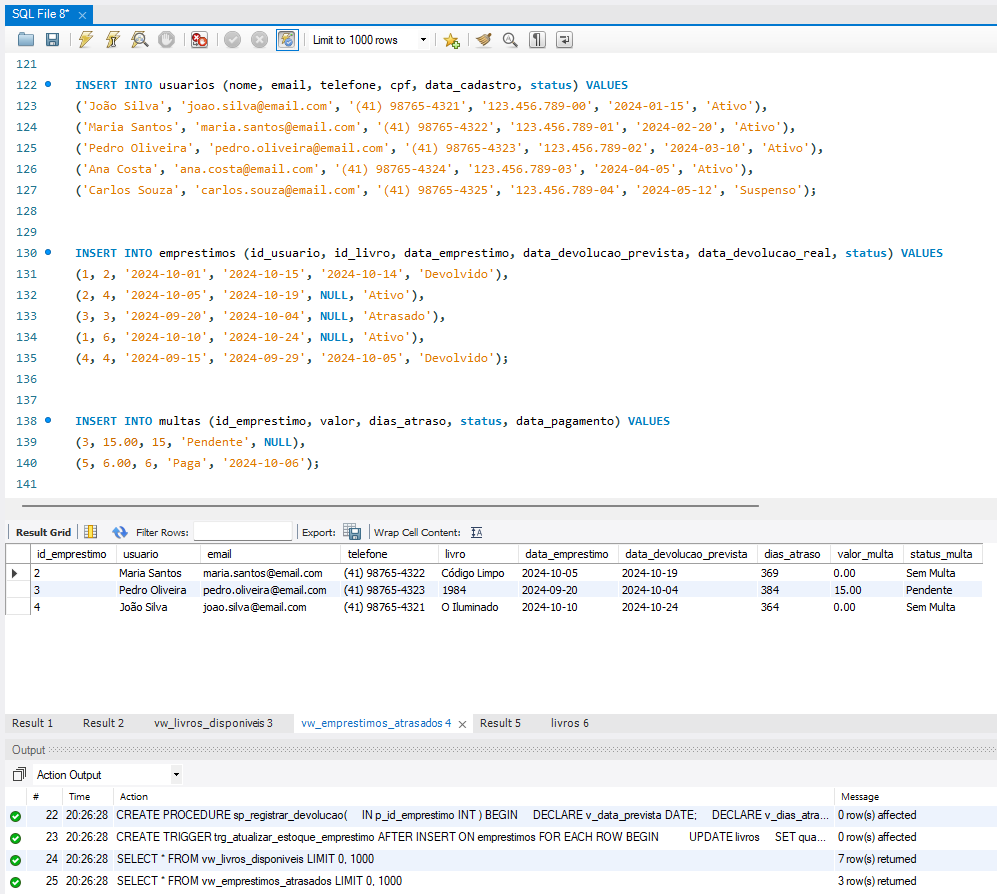
* id\_usuario (PK)
* nome
* email
* telefone
* cpf
* data\_cadastro
* status

**Tabela: emprestimos**

* id\_emprestimo (PK)
* id\_usuario (FK → usuarios)
* id\_livro (FK → livros)
* data\_emprestimo
* data\_devolucao\_prevista
* data\_devolucao\_real
* status

**Tabela: multas**

* id\_multa (PK)
* id\_emprestimo (FK → emprestimos)
* valor
* dias\_atraso
* status
* data\_pagamento



**3. DICIONÁRIO DE DADOS**

**3.1 Tabela: categorias**

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** | **Restrições** |
| id\_categoria | INT | Identificador único da categoria | PK, AUTO\_INCREMENT |
| nome | VARCHAR(100) | Nome da categoria | NOT NULL |
| descricao | TEXT | Descrição detalhada da categoria | - |

**3.2 Tabela: autores**

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** | **Restrições** |
| id\_autor | INT | Identificador único do autor | PK, AUTO\_INCREMENT |
| nome | VARCHAR(150) | Nome completo do autor | NOT NULL |
| nacionalidade | VARCHAR(50) | País de origem do autor | - |
| data\_nascimento | DATE | Data de nascimento | - |

**3.3 Tabela: livros**

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** | **Restrições** |
| id\_livro | INT | Identificador único do livro | PK, AUTO\_INCREMENT |
| titulo | VARCHAR(200) | Título da obra | NOT NULL |
| isbn | VARCHAR(20) | Código ISBN do livro | UNIQUE |
| ano\_publicacao | INT | Ano de publicação | - |
| quantidade\_total | INT | Quantidade total de exemplares | NOT NULL, DEFAULT 1 |
| quantidade\_disponivel | INT | Exemplares disponíveis | NOT NULL, DEFAULT 1 |
| id\_categoria | INT | Referência à categoria | FK |
| status | ENUM | Status do livro | 'Ativo', 'Inativo' |

**3.4 Tabela: livros\_autores**

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** | **Restrições** |
| id\_livro | INT | Referência ao livro | PK, FK |
| id\_autor | INT | Referência ao autor | PK, FK |

**3.5 Tabela: usuários**

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** | **Restrições** |
| id\_usuario | INT | Identificador único do usuário | PK, AUTO\_INCREMENT |
| nome | VARCHAR(150) | Nome completo do usuário | NOT NULL |
| email | VARCHAR(100) | E-mail do usuário | UNIQUE, NOT NULL |
| telefone | VARCHAR(20) | Telefone de contato | - |
| cpf | VARCHAR(14) | CPF do usuário | UNIQUE, NOT NULL |
| data\_cadastro | DATE | Data de cadastro no sistema | NOT NULL |
| status | ENUM | Status do usuário | 'Ativo', 'Inativo', 'Suspenso' |

**3.6 Tabela: empréstimos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** | **Restrições** |
| id\_emprestimo | INT | Identificador único do empréstimo | PK, AUTO\_INCREMENT |
| id\_usuario | INT | Referência ao usuário | FK, NOT NULL |
| id\_livro | INT | Referência ao livro | FK, NOT NULL |
| data\_emprestimo | DATE | Data do empréstimo | NOT NULL |
| data\_devolucao\_prevista | DATE | Data prevista para devolução | NOT NULL |
| data\_devolucao\_real | DATE | Data efetiva da devolução | - |
| status | ENUM | Status do empréstimo | 'Ativo', 'Devolvido', 'Atrasado' |

**3.7 Tabela: multas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** | **Restrições** |
| id\_multa | INT | Identificador único da multa | PK, AUTO\_INCREMENT |
| id\_emprestimo | INT | Referência ao empréstimo | FK, NOT NULL |
| valor | DECIMAL(10,2) | Valor da multa em reais | NOT NULL |
| dias\_atraso | INT | Quantidade de dias em atraso | NOT NULL |
| status | ENUM | Status da multa | 'Pendente', 'Paga' |
| data\_pagamento | DATE | Data do pagamento da multa | - |

**4. VIEWS (VISÕES)**

**4.1 vw\_livros\_disponiveis**

**Objetivo:** Fornecer uma visão consolidada de todos os livros disponíveis para empréstimo, incluindo informações sobre categoria e autores.

**Estrutura:**

|  |
| --- |
| sql |
| SELECT |
| l.id\_livro, |
| l.titulo, |
| l.isbn, |
| l.ano\_publicacao, |
| l.quantidade\_disponivel, |
| c.nome AS categoria, |
| GROUP\_CONCAT(a.nome SEPARATOR ', ') AS autores |
| FROM livros l |
| INNER JOIN categorias c ON l.id\_categoria = c.id\_categoria |
| LEFT JOIN livros\_autores la ON l.id\_livro = la.id\_livro |
| LEFT JOIN autores a ON la.id\_autor = a.id\_autor |
| WHERE l.quantidade\_disponivel > 0 AND l.status = 'Ativo' |
| GROUP BY l.id\_livro |

**Campos Retornados:**

* id\_livro: Identificador do livro
* titulo: Título da obra
* isbn: Código ISBN
* ano\_publicacao: Ano de publicação
* quantidade\_disponivel: Exemplares disponíveis
* categoria: Nome da categoria
* autores: Lista de autores (concatenados)

**Uso:** Consulta rápida para verificar quais livros estão disponíveis para empréstimo.

**4.2 vw\_emprestimos\_atrasados**

**Objetivo:** Listar todos os empréstimos em atraso com informações detalhadas do usuário, livro e multa associada.

**Estrutura:**

|  |
| --- |
| sql |
| SELECT |
| e.id\_emprestimo, |
| u.nome AS usuario, |
| u.email, |
| u.telefone, |
| l.titulo AS livro, |
| e.data\_emprestimo, |
| e.data\_devolucao\_prevista, |
| DATEDIFF(CURDATE(), e.data\_devolucao\_prevista) AS dias\_atraso, |
| COALESCE(m.valor, 0) AS valor\_multa, |
| COALESCE(m.status, 'Sem Multa') AS status\_multa |
| FROM emprestimos e |
| INNER JOIN usuarios u ON e.id\_usuario = u.id\_usuario |
| INNER JOIN livros l ON e.id\_livro = l.id\_livro |
| LEFT JOIN multas m ON e.id\_emprestimo = m.id\_emprestimo |
| WHERE e.status = 'Atrasado' OR (e.status = 'Ativo' AND e.data\_devolucao\_prevista < CURDATE()) |

**Campos Retornados:**

* id\_emprestimo: Identificador do empréstimo
* usuario: Nome do usuário
* email: E-mail para contato
* telefone: Telefone para contato
* livro: Título do livro emprestado
* data\_emprestimo: Data do empréstimo
* data\_devolucao\_prevista: Data prevista de devolução
* dias\_atraso: Quantidade de dias em atraso
* valor\_multa: Valor da multa calculada
* status\_multa: Status do pagamento da multa

**Uso:** Relatório gerencial para acompanhamento de devoluções atrasadas e cobrança de multas.

**5. CONSULTAS SQL COMPLEXAS**

**5.1 Consulta: Livros Mais Emprestados**

**Objetivo:** Identificar os livros mais populares da biblioteca baseado no histórico de empréstimos.

**Técnicas Utilizadas:**

* INNER JOIN: Para relacionar livros com categorias e empréstimos
* LEFT JOIN: Para incluir informações de autores
* GROUP BY: Para agrupar por livro
* COUNT: Para contar empréstimos
* ORDER BY: Para ordenar por popularidade

**Query:**

|  |
| --- |
| sql |
| SELECT |
| l.titulo, |
| c.nome AS categoria, |
| COUNT(e.id\_emprestimo) AS total\_emprestimos, |
| GROUP\_CONCAT(DISTINCT a.nome SEPARATOR ', ') AS autores |
| FROM livros l |
| INNER JOIN categorias c ON l.id\_categoria = c.id\_categoria |
| INNER JOIN emprestimos e ON l.id\_livro = e.id\_livro |
| LEFT JOIN livros\_autores la ON l.id\_livro = la.id\_livro |
| LEFT JOIN autores a ON la.id\_autor = a.id\_autor |
| GROUP BY l.id\_livro, l.titulo, c.nome |
| ORDER BY total\_emprestimos DESC |

**Resultado:** Lista ordenada dos livros mais emprestados com categoria e autores.

**5.2 Consulta: Usuários com Multas Pendentes**

**Objetivo:** Identificar usuários com pendências financeiras e suas respectivas dívidas.

**Técnicas Utilizadas:**

* Subqueries: Para calcular totais de empréstimos e multas
* IN: Para filtrar usuários com multas pendentes
* SUM: Para somar valores de multas
* INNER JOIN: Para relacionar tabelas

**Query:**

|  |
| --- |
| sql |
| SELECT |
| u.nome, |
| u.email, |
| u.status, |
| (SELECT COUNT(\*) |
| FROM emprestimos e |
| WHERE e.id\_usuario = u.id\_usuario) AS total\_emprestimos, |
| (SELECT SUM(m.valor) |
| FROM multas m |
| INNER JOIN emprestimos e ON m.id\_emprestimo = e.id\_emprestimo |
| WHERE e.id\_usuario = u.id\_usuario AND m.status = 'Pendente') AS total\_multas\_pendentes |
| FROM usuarios u |
| WHERE u.id\_usuario IN ( |
| SELECT DISTINCT e.id\_usuario |
| FROM emprestimos e |
| INNER JOIN multas m ON e.id\_emprestimo = m.id\_emprestimo |
| WHERE m.status = 'Pendente' |
| ) |
| ORDER BY total\_multas\_pendentes DESC |

**Resultado:** Lista de usuários com débitos pendentes ordenados por valor devido.

**6. STORED PROCEDURE**

**6.1 sp\_registrar\_devolucao**

**Objetivo:** Automatizar o processo de devolução de livros, incluindo atualização de estoque e cálculo automático de multas por atraso.

**Parâmetros:**

* IN p\_id\_emprestimo (INT): Identificador do empréstimo a ser finalizado

**Lógica de Funcionamento:**

1. **Busca de Informações**: Recupera a data prevista de devolução e o ID do livro
2. **Atualização do Empréstimo**: Define a data real de devolução e altera o status para 'Devolvido'
3. **Atualização de Estoque**: Incrementa a quantidade disponível do livro
4. **Cálculo de Atraso**: Calcula a diferença entre a data atual e a data prevista
5. **Geração de Multa**: Se houver atraso, cria registro de multa (R$ 1,00 por dia)
6. **Retorno de Mensagem**: Informa o resultado da operação

**Código:**

|  |
| --- |
| sql |
| CREATE PROCEDURE sp\_registrar\_devolucao( |
| IN p\_id\_emprestimo INT |
| ) |
| BEGIN |
| DECLARE v\_data\_prevista DATE; |
| DECLARE v\_dias\_atraso INT; |
| DECLARE v\_valor\_multa DECIMAL(10,2); |
| DECLARE v\_id\_livro INT; |
|  |
| *-- Buscar informações do empréstimo* |
| SELECT data\_devolucao\_prevista, id\_livro |
| INTO v\_data\_prevista, v\_id\_livro |
| FROM emprestimos |
| WHERE id\_emprestimo = p\_id\_emprestimo; |
|  |
| *-- Atualizar empréstimo* |
| UPDATE emprestimos |
| SET data\_devolucao\_real = CURDATE(), |
| status = 'Devolvido' |
| WHERE id\_emprestimo = p\_id\_emprestimo; |
|  |
| *-- Atualizar quantidade disponível do livro* |
| UPDATE livros |
| SET quantidade\_disponivel = quantidade\_disponivel + 1 |
| WHERE id\_livro = v\_id\_livro; |
|  |
| *-- Calcular dias de atraso* |
| SET v\_dias\_atraso = DATEDIFF(CURDATE(), v\_data\_prevista); |
|  |
| *-- Se houver atraso, criar multa* |
| IF v\_dias\_atraso > 0 THEN |
| SET v\_valor\_multa = v\_dias\_atraso \* 1.00; |
|  |
| INSERT INTO multas (id\_emprestimo, valor, dias\_atraso, status) |
| VALUES (p\_id\_emprestimo, v\_valor\_multa, v\_dias\_atraso, 'Pendente'); |
|  |
| SELECT CONCAT('Devolução registrada com multa de R$ ', v\_valor\_multa) AS mensagem; |
| ELSE |
| SELECT 'Devolução registrada sem multa' AS mensagem; |
| END IF; |
| END |

**Exemplo de Uso:**

|  |
| --- |
| sql |
| CALL sp\_registrar\_devolucao(3); |

**Benefícios:**

* Automatização do processo de devolução
* Cálculo automático de multas
* Atualização consistente de múltiplas tabelas
* Redução de erros humanos

**7. TRIGGER**

**7.1 trg\_atualizar\_estoque\_emprestimo**

**Objetivo:** Garantir a consistência do estoque de livros automaticamente quando um novo empréstimo é registrado.

**Tipo:** AFTER INSERT **Tabela:** emprestimos

**Lógica de Funcionamento:**

1. **Redução de Estoque**: Diminui a quantidade disponível do livro em 1 unidade
2. **Verificação de Disponibilidade**: Verifica se o livro ficou sem estoque
3. **Atualização de Status**: Se não houver mais exemplares disponíveis, marca o livro como 'Inativo'

**Código:**

|  |
| --- |
| sql |
| CREATE TRIGGER trg\_atualizar\_estoque\_emprestimo |
| AFTER INSERT ON emprestimos |
| FOR EACH ROW |
| BEGIN |
| *-- Diminuir quantidade disponível do livro* |
| UPDATE livros |
| SET quantidade\_disponivel = quantidade\_disponivel - 1 |
| WHERE id\_livro = NEW.id\_livro; |
|  |
| *-- Verificar se o livro ficou sem estoque* |
| UPDATE livros |
| SET status = 'Inativo' |
| WHERE id\_livro = NEW.id\_livro |
| AND quantidade\_disponivel = 0; |
| END |

**Características:**

* Execução automática e transparente
* Mantém integridade referencial
* Previne inconsistências no estoque
* Atualiza status automaticamente

**Impacto:** Toda vez que um novo empréstimo é inserido, o estoque é automaticamente atualizado.

**8. TESTES E VALIDAÇÕES**

**8.1 Teste das Views**

**Teste 1: vw\_livros\_disponiveis**

|  |
| --- |
| sql |
| SELECT \* FROM vw\_livros\_disponiveis; |

**Resultado Esperado:** Lista de todos os livros com quantidade disponível maior que zero.

**Teste 2: vw\_emprestimos\_atrasados**

|  |
| --- |
| sql |
| SELECT \* FROM vw\_emprestimos\_atrasados; |

**Resultado Esperado:** Lista de empréstimos com data de devolução vencida.

**8.2 Teste da Procedure**

|  |
| --- |
| sql |
| CALL sp\_registrar\_devolucao(3); |

**Validações:**

* Empréstimo deve ter status alterado para 'Devolvido'
* data\_devolucao\_real deve ser preenchida
* Quantidade disponível do livro deve aumentar
* Multa deve ser criada se houver atraso

**8.3 Teste do Trigger**

|  |
| --- |
| sql |
| INSERT INTO emprestimos (id\_usuario, id\_livro, data\_emprestimo, data\_devolucao\_prevista, status) |
| VALUES (2, 1, CURDATE(), DATE\_ADD(CURDATE(), INTERVAL 14 DAY), 'Ativo'); |
|  |
| SELECT titulo, quantidade\_disponivel FROM livros WHERE id\_livro = 1; |

**Validações:**

* Quantidade disponível deve diminuir em 1
* Se quantidade chegar a 0, status deve mudar para 'Inativo'

**9. REGRAS DE NEGÓCIO IMPLEMENTADAS**

**9.1 Gestão de Estoque**

* Quantidade total vs. quantidade disponível separadas
* Atualização automática ao emprestar e devolver
* Bloqueio automático quando sem estoque

**9.2 Controle de Multas**

* Valor fixo de R$ 1,00 por dia de atraso
* Cálculo automático na devolução
* Rastreamento de pagamento

**9.3 Status de Usuários**

* Ativo: Pode realizar empréstimos
* Suspenso: Não pode realizar novos empréstimos
* Inativo: Cadastro desativado

**9.4 Status de Empréstimos**

* Ativo: Empréstimo em andamento dentro do prazo
* Atrasado: Empréstimo com devolução vencida
* Devolvido: Empréstimo finalizado

**9.5 Prazo de Empréstimo**

* Prazo padrão: 7 dias
* Extensível conforme necessidade da biblioteca

**10. SEGURANÇA E INTEGRIDADE**

**10.1 Chaves Primárias**

Todas as tabelas possuem chaves primárias com AUTO\_INCREMENT para garantir unicidade.

**10.2 Chaves Estrangeiras**

Relacionamentos implementados com chaves estrangeiras garantindo integridade referencial.

**10.3 Constraints**

* NOT NULL em campos obrigatórios
* UNIQUE em campos que não podem duplicar (email, cpf, isbn)
* ENUM para campos com valores limitados

**10.4 Validações**

* Verificação de disponibilidade antes do empréstimo
* Validação de status de usuário
* Controle de estoque automático

**11. MELHORIAS FUTURAS**

**11.1 Funcionalidades Adicionais**

* Sistema de reservas
* Renovação automática de empréstimos
* Histórico completo de transações
* Relatórios estatísticos avançados
* Sistema de notificações por e-mail

**11.2 Otimizações**

* Índices adicionais para consultas frequentes
* Particionamento de tabelas grandes
* Cache de consultas comuns
* Arquivamento de dados históricos

**11.3 Segurança**

* Implementação de usuários e permissões
* Auditoria de operações
* Backup automático
* Criptografia de dados sensíveis

**12. CONCLUSÃO**

O Sistema de Gestão de Biblioteca apresentado atende a todos os requisitos propostos, implementando um banco de dados robusto e funcional com:

* ✅ Mais de 5 tabelas relacionadas
* ✅ 2 Views para consultas otimizadas
* ✅ 2 Consultas complexas com JOIN, GROUP BY e subquery
* ✅ 1 Stored Procedure para automação
* ✅ 1 Trigger para manutenção de integridade
* ✅ Dados de teste completos
* ✅ Modelo Conceitual e Lógico bem definidos

O sistema é escalável, seguro e preparado para uso em ambiente de produção com possibilidades de expansão futura.

**REFERÊNCIAS**

* MySQL Documentation: https://dev.mysql.com/doc/
* SQL Standards: ISO/IEC 9075
* Database Design Best Practices
* Modelagem de Dados: Peter Chen (1976)

**Data:** Outubro/2025  
**Versão:** 1.0  
**Desenvolvido para:** Trabalho Acadêmico - Banco de Dados