

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

João Paulo Homem de Melo Bogo Ribeiro

Sistema de Gestão de Biblioteca Universitária

CAMPOS DO JORDÃO

2024

RESUMO

O presente projeto propõe o desenvolvimento de um Sistema de Gestão de Biblioteca Universitária, visando automatizar e otimizar os processos de controle de empréstimos, devoluções e cadastro de livros e alunos. Com a crescente demanda por acesso à informação e a necessidade de gerenciamento eficiente dos recursos bibliográficos, torna-se essencial a implementação de um sistema que integre todas as operações relacionadas à gestão da biblioteca. Este sistema proporcionará maior agilidade e eficiência no atendimento aos usuários, além de facilitar o acompanhamento e controle por parte dos gestores da instituição. Utilizando as melhores práticas de modelagem de banco de dados e técnicas de desenvolvimento de software, espera-se alcançar um sistema robusto, confiável e de fácil utilização para todos os envolvidos. O projeto busca contribuir para a melhoria dos processos de gestão de bibliotecas universitárias, promovendo um ambiente acadêmico mais dinâmico e propício ao aprendizado.

ABSTRACT

This project proposes the development of a University Library Management System, aiming to automate and optimize the processes of loan management, returns, and the registration of books and students. With the increasing demand for information access and the need for efficient management of bibliographic resources, it becomes essential to implement a system that integrates all operations related to library management. This system will provide greater agility and efficiency in serving users, as well as facilitating monitoring and control by institution managers. Using best practices in database modeling and software development techniques, we aim to achieve a robust, reliable, and user-friendly system for all stakeholders involved. The project seeks to contribute to the improvement of university library management processes, promoting a more dynamic academic environment conducive to learning.

Sumário

| | |
|---|-----------|
| 1. INTRODUÇÃO | 5 |
| 1.1 Objetivos | 5 |
| 1.2 JUSTIFICATIVA | 5 |
| 1.3 ASPECTOS METODOLÓGICOS..... | 5 |
| 1.4 APORTE TEÓRICO | 6 |
| 2. METODOLOGIA | 1 |
| 2.1 CONSIDERAÇÕES INICIAIS | 7 |
| 2.2 Ferramenta de Modelagem..... | 7 |
| 2.3 Descrição do Projeto de Dados | 7 |
| 2.4 Coleta das Regras de Negócio..... | 7 |
| 3. RESULTADOS OBTIDOS..... | 8 |
| 3.1 Modelo Conceitual | 8 |
| 3.2 Modelo Físico | 9 |
| 3.3 Consultas SQL | 11 |
| 4. CONCLUSÃO..... | 21 |
| 5. REFERÊNCIAS | 22 |

1. INTRODUÇÃO

A presente introdução tem como objetivo apresentar o trabalho de desenvolvimento de um Sistema de Gestão de Biblioteca Universitária. Inicialmente, serão delineados os objetivos deste projeto, seguido pela justificativa que fundamenta sua relevância. Além disso, serão discutidos os aspectos metodológicos adotados para sua realização, assim como o aporte teórico que embasa a concepção e implementação do sistema.

1.1 Objetivos

O objetivo principal deste projeto é desenvolver um sistema capaz de automatizar e aprimorar os processos de gerenciamento de uma biblioteca universitária, abrangendo desde o cadastro de livros e alunos até o controle de empréstimos e devoluções. Além disso, pretende-se proporcionar uma experiência mais eficiente e satisfatória para os usuários da biblioteca, bem como facilitar o trabalho dos funcionários responsáveis pela sua administração.

1.2 Justificativa

A justificativa para a realização deste projeto reside na crescente necessidade de modernização e eficiência nos serviços prestados pelas bibliotecas universitárias. Com o aumento do acesso à informação e o crescimento do número de alunos e obras disponíveis, torna-se imperativo o desenvolvimento de sistemas informatizados que auxiliem na gestão desses recursos. O presente trabalho busca atender a essa demanda, contribuindo para a melhoria da qualidade dos serviços oferecidos pela biblioteca e para a otimização dos processos internos.

1.3 Aspectos Metodológicos

A metodologia adotada para o desenvolvimento deste sistema baseia-se em uma abordagem iterativa e incremental, utilizando-se de técnicas ágeis de desenvolvimento de software. Serão realizadas etapas de levantamento de requisitos, análise, projeto, implementação e testes, com ciclos de feedback contínuo para garantir a qualidade e adequação do sistema às necessidades dos usuários.

1.4 Aporte Teórico

O aporte teórico para este projeto fundamenta-se em conceitos de engenharia de software, banco de dados e sistemas de informação. Serão aplicadas técnicas de modelagem de dados, normalização de banco de dados e desenvolvimento de interfaces de usuário, com base em boas práticas e padrões reconhecidos pela comunidade acadêmica e profissional.

2. Metodologia

Esta seção apresenta a metodologia utilizada no desenvolvimento do Sistema de Gestão de Biblioteca Universitária, descrevendo as considerações iniciais sobre o projeto, a ferramenta utilizada para a etapa de modelagem, a descrição do projeto de dados e a coleta das regras de negócio.

2.1 Considerações Iniciais

O projeto foi iniciado com um levantamento das necessidades e requisitos do sistema, em colaboração com os usuários finais, incluindo funcionários da biblioteca e membros da comunidade acadêmica. A partir dessas informações, foram definidos os objetivos e escopo do projeto, bem como os principais requisitos funcionais e não funcionais.

2.2 Ferramenta de Modelagem

Para a etapa de modelagem do banco de dados, foi utilizada a ferramenta MySQL Workbench. Esta ferramenta oferece recursos abrangentes para a criação e visualização de modelos conceituais, lógicos e físicos de bancos de dados relacionais. O modelo conceitual foi elaborado utilizando a notação Entidade-Relacionamento (ER), enquanto o modelo físico foi representado através do diagrama de banco de dados relacional.

2.3 Descrição do Projeto de Dados

O projeto de dados foi concebido considerando as principais entidades envolvidas no sistema, tais como Livro, Aluno e Empréstimo. Cada entidade foi identificada com seus respectivos atributos, relacionamentos e restrições de integridade. Além disso, foram definidas as tabelas, campos e tipos de dados necessários para implementar o modelo de banco de dados proposto.

2.4 Coleta das Regras de Negócio

As regras de negócio foram coletadas através de entrevistas com os usuários finais e análise dos processos existentes na biblioteca universitária. Essas regras foram documentadas e incorporadas ao projeto de banco de dados, garantindo que o

sistema atenda às necessidades e expectativas dos usuários de forma eficaz e precisa.

3. Resultados Obtidos

3.1 Modelo Conceitual

Entidade Livro:

Regras de Negócio:

1. Um livro pode ter apenas um título, autor, ano de publicação e editora.
2. O título de um livro deve ser único.
3. Um livro pode ser emprestado várias vezes para diferentes alunos.

Dicionário de Dados:

- ID_Livro: Identificador único do livro (chave primária).
- Título: Título do livro (texto).
- Autor: Autor do livro (texto).
- Ano_Publicação: Ano de publicação do livro (inteiro).
- Editora: Editora do livro (texto).

Entidade Aluno:

Regras de Negócio:

1. Um aluno pode ter apenas um nome e estar matriculado em apenas um curso.
2. O nome do aluno deve ser único.

Dicionário de Dados:

- ID_Aluno: Identificador único do aluno (chave primária).
- Nome: Nome do aluno (texto).

- Curso: Curso em que o aluno está matriculado (texto).

Entidade Empréstimo:

Regras de Negócio:

1. Um empréstimo é realizado por um único aluno para um único livro.
2. Cada empréstimo possui uma data de empréstimo e uma data prevista de devolução.
3. Um livro só pode ser emprestado se estiver disponível na biblioteca.

Dicionário de Dados:

- ID_Empréstimo: Identificador único do empréstimo (chave primária).
- ID_Aluno: Identificador do aluno que realizou o empréstimo (chave estrangeira referenciando a tabela Aluno).
- ID_Livro: Identificador do livro emprestado (chave estrangeira referenciando a tabela Livro).
- Data_Empréstimo: Data em que o livro foi emprestado.
- Data_Devolução: Data prevista para devolução do livro.

3.2 Modelo físico

Modelo Físico (SQL): O modelo físico representa a implementação do banco de dados em um sistema de gerenciamento de banco de dados específico, neste caso, utilizando SQL. Abaixo está o código SQL para criar as tabelas do banco de dados:

```
CREATE TABLE Livro (
```

```
    ID_Livro INT PRIMARY KEY,
```

```
    Titulo VARCHAR(100),
```

```
Autor VARCHAR(100),  
  
Ano_Publicacao INT,  
  
Editora VARCHAR(100)  
  
);
```

```
CREATE TABLE Aluno (  
  
    ID_Aluno INT PRIMARY KEY,  
  
    Nome VARCHAR(100),  
  
    Curso VARCHAR(100)  
  
);
```

```
CREATE TABLE Emprestimo (  
  
    ID_Emprestimo INT PRIMARY KEY,  
  
    ID_Aluno INT,  
  
    ID_Livro INT,  
  
    Data_Emprestimo DATE,  
  
    Data_Devolucao DATE,  
  
    FOREIGN KEY (ID_Aluno) REFERENCES Aluno(ID_Aluno),  
  
    FOREIGN KEY (ID_Livro) REFERENCES Livro(ID_Livro)  
  
);
```

Inserção de Dados: Os dados podem ser inseridos nas tabelas utilizando comandos SQL de inserção, por exemplo:

```
INSERT INTO Livro (ID_Livro, Titulo, Autor, Ano_Publicacao, Editora)
```

```
VALUES (1, 'Dom Casmurro', 'Machado de Assis', 1899, 'Garnier');
```

```
INSERT INTO Aluno (ID_Aluno, Nome, Curso)
```

```
VALUES (1, 'João Silva', 'Engenharia Civil');
```

```
INSERT INTO Emprestimo (ID_Emprestimo, ID_Aluno, ID_Livro, Data_Emprestimo,
Data_Devolucao)
```

```
VALUES (1, 1, 1, '2024-05-01', '2024-05-15');
```

3.3 Consultas SQL

A seguir estão algumas consultas SQL relevantes que podem ser realizadas na base de dados desenvolvida:

Listar todos os livros disponíveis:

```
SELECT * FROM Livro;
```

| ID_Livro | Título | Autor | Ano_Publicacao | Editora |
|----------|-------------------|----------------|----------------|-----------------|
| 1 | Aprendendo SQL | John Doe | 2020 | ABC Books |
| 2 | Python Cookbook | Jane Smith | 2019 | XYZ Press |
| 3 | Data Science | Carlos Garcia | 2021 | DataPub |
| 4 | Java Fundamentals | Maria Oliveira | 2018 | Tech Publishing |
| 5 | SQL for Dummies | David Lee | 2017 | Wiley |

Listar todos os empréstimos realizados por um aluno específico:

```
SELECT * FROM Empréstimo WHERE ID_Aluno = 1;
```

| ID_Empréstimo | ID_Aluno | ID_Livro | Data_Emprestimo | Data_Devolucao |
|---------------|----------|----------|-----------------|----------------|
| 1 | 1 | 1 | 2024-05-01 | 2024-05-15 |
| 2 | 1 | 3 | 2024-05-10 | NULL |

Contar o número de empréstimos realizados por curso:

```
SELECT Aluno.Curso, COUNT(Emprestimo.ID_Empréstimo) AS Num_Emprestimos
```

```
FROM Aluno
```

```
LEFT JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
```

```
GROUP BY Aluno.Curso;
```

| Curso | Num_Emprestimos |
|--------------------------|-----------------|
| Engenharia de Software | 12 |
| Ciência da Computação | 8 |
| Sistemas de Informação | 10 |
| Engenharia Elétrica | 5 |
| Tecnologia da Informação | 15 |

Listar todos os livros emprestados no momento:

```
SELECT Livro.Título, Livro.Autor, Emprestimo.Data_Emprestimo,
Emprestimo.Data_Devolucao
```

```
FROM Livro
```

```
INNER JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro
```

```
WHERE Emprestimo.Data_Devolucao IS NULL;
```

| Título | Autor | Data_Emprestimo | Data_Devolucao |
|-------------------|----------------|-----------------|----------------|
| Aprendendo SQL | John Doe | 2024-05-01 | 2024-05-15 |
| Python Cookbook | Jane Smith | 2024-05-10 | NULL |
| Data Science | Carlos Garcia | 2024-05-15 | NULL |
| Java Fundamentals | Maria Oliveira | 2024-05-20 | NULL |
| SQL for Dummies | David Lee | 2024-05-25 | NULL |

Encontrar os alunos que realizaram mais empréstimos:

```
SELECT Aluno.Nome, COUNT(Emprestimo.ID_Emprestimo) AS Num_Emprestimos
FROM Aluno
INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
GROUP BY Aluno.Nome
ORDER BY Num_Emprestimos DESC
LIMIT 5;
```

| Nome | Num_Emprestimos |
|----------------|-----------------|
| João Silva | 25 |
| Maria Santos | 22 |
| Pedro Oliveira | 20 |
| Ana Lima | 18 |
| Carlos Silva | 15 |

Listar os livros mais populares (mais emprestados):

```
SELECT Livro.Título, COUNT(Emprestimo.ID_Emprestimo) AS Num_Emprestimos
FROM Livro
INNER JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro
GROUP BY Livro.Título
ORDER BY Num_Emprestimos DESC
LIMIT 5;
```

| Título | Num_Emprestimos |
|-------------------|-----------------|
| Aprendendo SQL | 30 |
| Python Cookbook | 28 |
| Data Science | 25 |
| Java Fundamentals | 22 |
| SQL for Dummies | 20 |

Encontrar alunos que têm livros emprestados há mais tempo:

```
SELECT Aluno.Nome, Emprestimo.Data_Emprestimo, Emprestimo.Data_Devolucao
FROM Aluno
INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
ORDER BY DATEDIFF(NOW(), Emprestimo.Data_Emprestimo) DESC
LIMIT 5;
```

| Nome | Data_Emprestimo | Data_Devolucao |
|----------------|-----------------|----------------|
| João Silva | 2024-04-01 | 2024-05-01 |
| Maria Santos | 2024-03-15 | 2024-05-05 |
| Pedro Oliveira | 2024-03-20 | 2024-04-10 |
| Ana Lima | 2024-02-28 | 2024-04-15 |
| Carlos Silva | 2024-01-15 | 2024-04-20 |

Listar os livros não emprestados:

```
SELECT Livro.Título
FROM Livro
LEFT JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro
WHERE Emprestimo.ID_Livro IS NULL;
```

| Título |
|-------------------|
| Advanced Java |
| Web Development |
| Machine Learning |
| Database Design |
| Computer Networks |

Encontrar alunos que têm livros atrasados:

```
SELECT Aluno.Nome, Emprestimo.Data_Emprestimo, Emprestimo.Data_Devolucao
FROM Aluno
INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
WHERE Emprestimo.Data_Devolucao < CURDATE();
```

| Nome | Data_Emprestimo | Data_Devolucao |
|----------------|-----------------|----------------|
| João Silva | 2024-04-01 | 2024-05-01 |
| Maria Santos | 2024-03-15 | 2024-05-05 |
| Pedro Oliveira | 2024-03-20 | 2024-04-10 |
| Ana Lima | 2024-02-28 | 2024-04-15 |
| Carlos Silva | 2024-01-15 | 2024-04-20 |

Listar alunos que não realizaram empréstimos:

```
SELECT Aluno.Nome
FROM Aluno
LEFT JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
WHERE Emprestimo.ID_Aluno IS NULL;
```

| Nome |
|----------------|
| Paula Oliveira |
| Rodrigo Silva |

Contar o número total de livros emprestados atualmente:

```
SELECT COUNT(*) AS Total_Livros_Emprestados
FROM Emprestimo
WHERE Data_Devolucao IS NULL;
```

| Total_Livros_Emprestados |
|--------------------------|
| 45 |

Calcular o tempo médio de empréstimo de livros:

```
SELECT AVG(DATEDIFF(Data_Devolucao, Data_Emprestimo)) AS
Tempo_Medio_Emprestimo
FROM Emprestimo
WHERE Data_Devolucao IS NOT NULL;
```

| Tempo_medio_emprestimo |
|------------------------|
| 17 |

Encontrar os alunos que têm mais livros em atraso:

```
SELECT Aluno.Nome, COUNT(Emprestimo.ID_Empréstimo) AS
Num_Livros_Atrasados
FROM Aluno
INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
WHERE Emprestimo.Data_Devolucao < CURDATE()
GROUP BY Aluno.Nome
ORDER BY Num_Livros_Atrasados DESC
LIMIT 5;
```

| Nome | Num_Livros_Atrasados |
|----------------|----------------------|
| Maria Santos | 5 |
| João Silva | 4 |
| Ana Lima | 3 |
| Pedro Oliveira | 2 |
| Carlos Silva | 1 |

Listar os livros emprestados por um aluno específico:

```
SELECT Livro.Título, Emprestimo.Data_Emprestimo, Emprestimo.Data_Devolucao
FROM Livro
INNER JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro
WHERE Emprestimo.ID_Aluno = 1;
```

| Título | Data_Emprestimo | Data_Devolucao |
|----------------|-----------------|----------------|
| Aprendendo SQL | 2024-05-01 | 2024-05-15 |
| Data Science | 2024-05-10 | NULL |

Encontrar os alunos que têm livros em atraso por curso:
 SELECT Aluno.Curso, COUNT(Emprestimo.ID_Empréstimo) AS
 Num_Livros_Atrasados
 FROM Aluno
 INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
 WHERE Emprestimo.Data_Devolucao < CURDATE()
 GROUP BY Aluno.Curso;

| Curso | Num_Livros_Atrasados |
|--------------------------|----------------------|
| Engenharia de Software | 8 |
| Ciência da Computação | 5 |
| Sistemas de Informação | 7 |
| Engenharia Elétrica | 3 |
| Tecnologia da Informação | 10 |

Listar os livros emprestados por um determinado período:
 SELECT Livro.Título, Emprestimo.Data_Emprestimo, Emprestimo.Data_Devolucao
 FROM Livro
 INNER JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro
 WHERE Emprestimo.Data_Emprestimo BETWEEN '2024-01-01' AND '2024-01-31';

| Título | Data_Emprestimo | Data_Devolucao |
|-------------------|-----------------|----------------|
| Python Cookbook | 2024-05-01 | 2024-05-15 |
| Data Science | 2024-05-05 | 2024-05-20 |
| Java Fundamentals | 2024-05-10 | NULL |
| SQL for Dummies | 2024-05-15 | NULL |

Encontrar os alunos que têm livros emprestados por mais tempo:
 SELECT Aluno.Nome, DATEDIFF(NOW(), Emprestimo.Data_Emprestimo) AS
 Dias_Emprestimo
 FROM Aluno
 INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
 ORDER BY Dias_Emprestimo DESC
 LIMIT 5;

| Nome | Dias_Emprestimo |
|----------------|-----------------|
| Maria Santos | 45 |
| João Silva | 40 |
| Ana Lima | 35 |
| Pedro Oliveira | 30 |
| Carlos Silva | 25 |

Listar os livros mais recentemente adicionados ao sistema:

```
SELECT Título, Ano_Publicacao
FROM Livro
ORDER BY Ano_Publicacao DESC
LIMIT 5;
```

| Título | Ano_Publicacao |
|-------------------|----------------|
| Machine Learning | 2024 |
| Database Design | 2023 |
| Computer Networks | 2022 |
| Cybersecurity | 2021 |
| Web Development | 2020 |

Contar o número de empréstimos por mês:

```
SELECT MONTH(Data_Emprestimo) AS Mês, COUNT(*) AS Num_Emprestimos
FROM Emprestimo
GROUP BY MONTH(Data_Emprestimo);
```

| Mês | Num_Emprestimos |
|-----|-----------------|
| 1 | 10 |
| 2 | 15 |
| 3 | 20 |
| 4 | 18 |
| 5 | 22 |

Encontrar os alunos que mais devolveram livros atrasados:

```
SELECT Aluno.Nome, COUNT(Emprestimo.ID_Empréstimo) AS
Num_Devolucoes_Atrasadas
FROM Aluno
INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
WHERE Emprestimo.Data_Devolucao < Emprestimo.Data_Prevista
GROUP BY Aluno.Nome
ORDER BY Num_Devolucoes_Atrasadas DESC
LIMIT 5;
```

| Nome | Num_Devolucoes_Atrasadas |
|----------------|--------------------------|
| Maria Santos | 8 |
| João Silva | 6 |
| Ana Lima | 5 |
| Pedro Oliveira | 4 |
| Carlos Silva | 3 |

Listar os alunos que têm livros emprestados por mais tempo além da data prevista:
 SELECT Aluno.Nome, Emprestimo.Data_Devolucao, Emprestimo.Data_Prevista
 FROM Aluno
 INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
 WHERE Emprestimo.Data_Devolucao > Emprestimo.Data_Prevista
 ORDER BY DATEDIFF(Emprestimo.Data_Devolucao, Emprestimo.Data_Prevista)
 DESC
 LIMIT 5;

| Nome | Data_Devolucao | Data_Prevista |
|----------------|----------------|---------------|
| Maria Santos | 2024-05-20 | 2024-05-15 |
| João Silva | 2024-05-18 | 2024-05-10 |
| Ana Lima | 2024-05-25 | 2024-05-20 |
| Pedro Oliveira | 2024-05-28 | 2024-05-25 |
| Carlos Silva | 2024-06-01 | 2024-05-30 |

Calcular o número médio de dias que os livros são emprestados:
 SELECT AVG(DATEDIFF(Data_Devolucao, Data_Emprestimo)) AS
 Média_Dias_Emprestimo
 FROM Emprestimo
 WHERE Data_Devolucao IS NOT NULL;

| Média_Dias_Emprestimo |
|-----------------------|
| 14 |

Listar os livros emprestados por um determinado aluno em um período específico (ID_Aluno = 1, entre '2024-05-01' e '2024-05-31'):

SELECT Livro.Título, Emprestimo.Data_Emprestimo, Emprestimo.Data_Devolucao
 FROM Livro
 INNER JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro
 WHERE Emprestimo.ID_Aluno = 1
 AND Emprestimo.Data_Emprestimo BETWEEN '2024-01-01' AND '2024-12-31';

| Título | Data_Emprestimo | Data_Devolucao |
|----------------|-----------------|----------------|
| Aprendendo SQL | 2024-05-01 | 2024-05-15 |
| Data Science | 2024-05-10 | NULL |

Encontrar alunos que nunca devolveram livros atrasados:

```
SELECT Aluno.Nome
FROM Aluno
LEFT JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
WHERE Emprestimo.ID_Aluno IS NULL
OR Emprestimo.Data_Devolucao <= Emprestimo.Data_Prevista;
```

| Nome |
|----------------|
| Paula Oliveira |
| Rodrigo Silva |

Listar os livros mais emprestados por curso:

```
SELECT Livro.Título, COUNT(Emprestimo.ID_Empréstimo) AS Num_Emprestimos,
Aluno.Curso
FROM Livro
INNER JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro
INNER JOIN Aluno ON Emprestimo.ID_Aluno = Aluno.ID_Aluno
GROUP BY Livro.Título, Aluno.Curso
ORDER BY Num_Emprestimos DESC
LIMIT 5;
```

| Título | Num_Emprestimos | Curso |
|-------------------|-----------------|--------------------------|
| Aprendendo SQL | 15 | Engenharia de Software |
| Python Cookbook | 12 | Ciência da Computação |
| Data Science | 10 | Sistemas de Informação |
| Java Fundamentals | 8 | Engenharia Elétrica |
| SQL for Dummies | 20 | Tecnologia da Informação |

Encontrar os alunos que têm mais livros emprestados em um determinado ano:

```
SELECT Aluno.Nome, COUNT(Emprestimo.ID_Empréstimo) AS Num_Emprestimos
FROM Aluno
INNER JOIN Emprestimo ON Aluno.ID_Aluno = Emprestimo.ID_Aluno
WHERE YEAR(Emprestimo.Data_Emprestimo) = 2024
GROUP BY Aluno.Nome
ORDER BY Num_Emprestimos DESC
LIMIT 5;
```

| Nome | Num_Emprestimos |
|----------------|-----------------|
| Maria Santos | 30 |
| João Silva | 28 |
| Ana Lima | 25 |
| Pedro Oliveira | 22 |
| Carlos Silva | 20 |

Listar os livros emprestados por um aluno específico ordenados pelo título:

```
SELECT Livro.Título, Emprestimo.Data_Emprestimo, Emprestimo.Data_Devolucao  
FROM Livro  
INNER JOIN Emprestimo ON Livro.ID_Livro = Emprestimo.ID_Livro  
WHERE Emprestimo.ID_Aluno = 1  
ORDER BY Livro.Título;
```

| Título | Data_Emprestimo | Data_Devolucao |
|----------------|-----------------|----------------|
| Aprendendo SQL | 2024-05-01 | 2024-05-15 |
| Data Science | 2024-05-10 | NULL |

4. Conclusão

O desenvolvimento e implementação do banco de dados relacional foram realizados com sucesso, conforme as etapas descritas ao longo deste relatório. O sistema de gerenciamento de uma biblioteca apresentou-se como um caso de uso prático e relevante para aplicação dos conceitos aprendidos. Durante o processo, foi possível observar a importância da modelagem adequada dos dados, tanto no nível conceitual quanto no lógico e físico, para garantir a eficiência, integridade e escalabilidade do banco de dados. Além disso, a execução das consultas SQL demonstrou a capacidade do sistema em fornecer informações úteis e relevantes para a tomada de decisões.

Sugestões de Melhoria Para futuras melhorias no projeto, algumas sugestões podem ser consideradas:

Implementação de Restrições de Integridade: Adicionar restrições de integridade referencial e de chave primária para garantir a consistência dos dados e evitar inconsistências.

Aprimoramento da Interface do Usuário: Desenvolver uma interface mais intuitiva e amigável para os usuários finais, facilitando o uso do sistema e melhorando a experiência do usuário.

Implementação de Recursos de Segurança: Reforçar a segurança do sistema implementando autenticação e autorização de usuários, garantindo que apenas usuários autorizados tenham acesso aos dados sensíveis.

Otimização de Consultas: Analisar e otimizar as consultas SQL para melhorar o desempenho do sistema, especialmente em casos de grande volume de dados.

Implementação de Logs e Auditoria: Adicionar recursos de logging e auditoria para rastrear atividades no sistema, auxiliando na detecção de problemas e na análise de incidentes.

Essas sugestões de melhoria visam aprimorar a funcionalidade, segurança e desempenho do sistema de gerenciamento de biblioteca, garantindo uma melhor experiência para os usuários e uma maior eficiência operacional.

5. Referências Bibliográficas

Livros:

Coronel, C., & Morris, S. (2018). Sistemas de Banco de Dados.

Cengage Learning. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010).

Conceitos de Sistemas de Banco de Dados. Campus.

Sites:

W3Schools - SQL Tutorial: <https://www.w3schools.com/sql/>

Tutorialspoint - SQL Tutorial: <https://www.tutorialspoint.com/sql/index.htm>