

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

**VINÍCIUS SOUSA LUCENA CHINAQUI**

**Sistema de Gestão de Academia.**

**CAMPOS DO JORDÃO  
2025**

## RESUMO

Este projeto apresenta o desenvolvimento de um banco de dados relacional para um **Sistema de Gestão de Academia**, com o objetivo de otimizar a organização e o controle das informações operacionais de academias. A proposta contempla funcionalidades como cadastro de alunos, controle de planos, mensalidades, frequência e histórico de treinos. O projeto foi desenvolvido com base na modelagem conceitual e lógica, aplicando os princípios da normalização de dados e boas práticas de design de banco de dados. A implementação foi realizada utilizando o Sistema Gerenciador de Banco de Dados MySQL, com estruturação em linguagem SQL. A metodologia empregada envolveu o levantamento de requisitos, criação de diagramas entidade-relacionamento (ER), definição de restrições de integridade e testes com dados simulados. O projeto visa oferecer uma base sólida e escalável para o desenvolvimento completo do sistema, promovendo maior eficiência, segurança e confiabilidade na gestão das academias.

**Palavras-chave:** Sistema de Gestão de Academia; Banco de Dados Relacional; MySQL; SQL; Modelagem de Dados

## **ABSTRACT**

This project presents the development of a relational database for a Gym Management System, aimed at optimizing the organization and control of operational information in fitness centers. The proposed solution includes features such as student registration, membership plan management, payment tracking, attendance control, and workout history. The project was developed based on conceptual and logical modeling, applying normalization principles and best practices in database design. Implementation was carried out using the MySQL Database Management System, structured in SQL. The adopted methodology involved requirements gathering, creation of entity-relationship (ER) diagrams, definition of integrity constraints, and testing with simulated data. The project aims to provide a solid and scalable foundation for the complete development of the system, promoting greater efficiency, security, and reliability in gym management.

**Keywords:** Gym Management System; Relational Database; MySQL; SQL; Data Modeling.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>6</b>
<b>1.1</b>	<b>Objetivos</b>	<b>6</b>
<b>1.2</b>	<b>Justificativa</b>	<b>7</b>
<b>1.3</b>	<b>Aspectos Metodológicos</b>	<b>7</b>
<b>1.4</b>	<b>Aporte Teórico</b>	<b>8</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>9</b>
<b>2.1</b>	<b>Banco de Dados Relacional</b>	<b>9</b>
<b>2.2</b>	<b>Modelagem de Dados e Normalização</b>	<b>10</b>
<b>2.3</b>	<b>Trabalhos Relacionados</b>	<b>10</b>
<b>3</b>	<b>PROJETO PROPOSTO (METODOLOGIA)</b>	<b>11</b>
<b>3.1</b>	<b>Apresentação de Figuras</b>	<b>11</b>
<b>3.2</b>	<b>Descrição de Entidades</b>	<b>13</b>
<b>3.3</b>	<b>Descrição de Relacionamentos</b>	<b>13</b>
<b>3.4</b>	<b>Apresentação de Algoritmos</b>	<b>16</b>
<b>3.5</b>	<b>Resultados</b>	<b>20</b>
<b>3.5.1</b>	<b>Modelo Conceitual</b>	<b>20</b>
<b>3.5.2</b>	<b>Dicionário de Dados</b>	<b>22</b>
<b>3.5.3</b>	<b>Consultas SQL Desenvolvidas</b>	<b>22</b>
<b>4</b>	<b>Conclusão</b>	<b>25</b>
<b>5</b>	<b>Referências</b>	<b>26</b>

## 1 INTRODUÇÃO

Nesse capítulo, se mostra os pontos chave abordados no desenvolver do projeto Sistema de Gestão de Academia. O projeto tem como meta criar um banco de dados relacional pra atender as precisar operacionais duma academia, permitindo a organização e o controle das informações como cadastro dos alunos, gestão de planos, controle dos pagamentos, registro de frequência e acompanhamento dos treinos.

Hoje em dia, muitas academias pequenas e de médio porte encaram desafios na gestão dos dados, usando planilhas ou processos manuais que podem gerar erros, retrabalho e dificuldades na hora de tomar decisões. Nesse cenário, aparece a precisão de desenvolver uma solução eficiente, estruturada e que consiga garantir a integridade e segurança dos dados.

O projeto foi desenvolvido usando uma metodologia que inclui o levantamento de requisitos, modelagem conceitual por meio de diagrama entidade-relacionamento, aplicação das regras de normalização, criando o modelo lógico e físico, além da implementação no Sistema Gerenciador de Banco de Dados MySQL.

### 1.1 Objetivos

O presente projeto tem por objetivo desenvolver um banco de dados relacional para um Sistema de Gestão de Academia, que possibilite o armazenamento, organização e gerenciamento eficiente das informações relacionadas às atividades administrativas de uma academia.

Para a consecução deste objetivo foram estabelecidos os objetivos específicos:

- Realizar o levantamento de requisitos e entender as regras de negócio aplicáveis à gestão de uma academia;
- Elaborar o modelo conceitual do banco de dados utilizando notação de Diagrama Entidade-Relacionamento (DER);
- Construir o modelo lógico e físico do banco de dados;
- Implementar o banco de dados no Sistema Gerenciador de Banco de Dados (SGBD) MySQL;

- Garantir a integridade dos dados por meio da aplicação de regras de normalização e integridade referencial;
- Gerar consultas SQL que atendam às necessidades de informação da academia.

## 1.2 Justificativa

A escolha por criar um sistema pra gerir academias surge da demanda cada vez maior por tecnologia que melhore os processos administrativos nesses lugares. É notório, muitas academias pequenas e médias ainda usam controles em papel ou planilhas, o que atrapalha a segurança, integridade, e facilidade de acesso as informações.

Diante disso, este projeto se justifica por oferecer uma solução bem feita, que torna a gestão mais eficiente, diminuindo erros, acelerando o acesso a informação, e criando um bom banco de dados pra futuras expansões do sistema, tipo integração com sistemas web, apps, ou geração de relatórios de gestão.

Além disso, o projeto ajuda na formação academica, pois permite aplicar na prática conceitos teóricos de banco de dados, modelagem, normalização e desenvolvimento de sistemas.

## 1.3 Aspectos Metodológicos

O desenvolvimento deste projeto foi conduzido por meio de uma abordagem prática, baseada em pesquisa aplicada e desenvolvimento técnico.

As etapas do projeto seguiram a seguinte sequência metodológica:

1. **Levantamento de Requisitos:** Definição das necessidades da academia e entendimento das regras de negócio;

2. **Modelagem de Dados:** Criação do modelo conceitual utilizando o Diagrama Entidade-Relacionamento (DER);
3. **Projeto Lógico:** Conversão do modelo conceitual para o modelo lógico relacional;
4. **Normalização:** Aplicação das regras de normalização até a terceira forma normal (3FN) para garantir a integridade dos dados;
5. **Projeto Físico:** Implementação do modelo no ambiente do SGBD MySQL;
6. **Testes:** Inserção de dados simulados e execução de consultas SQL para validação do projeto.

A coleta das informações sobre as regras de negócio foi realizada por meio de pesquisa em fontes secundárias, análise de funcionamento real de academias e referências teóricas sobre modelagem de banco de dados.

## 1.4 Aporte Teórico

Aqui são apresentados rapidamente as bases teóricas em que esta pesquisa se fundamentou. Depois haverá uma seção (2) na qual cada conceito e trabalho relacionado a este será explicitado detalhadamente.

Na seção 2 são apresentadas as principais bases teóricas que fundamentaram este trabalho. A seção 3 apresenta a metodologia de desenvolvimento e o sistema proposto neste trabalho. Na seção 4 é apresentada a avaliação do sistema desenvolvido. Por fim, a seção 5 apresenta a conclusão deste trabalho.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Nesta seção será apresentada uma revisão de textos, livros e materiais acadêmicos pertinentes à compreensão dos conceitos necessários para o desenvolvimento do Sistema de Gestão de Academia. Serão abordados temas relacionados a banco de dados, modelagem de dados e sistemas de gestão.

### **2.1 Banco de Dados Relacional**

O conceito de banco de dados relacional, surgido com Edgar Frank Codd lá por volta de 1970, propôs uma forma de organizar informações em tabelas bidimensionais, chamadas relações. Os dados seriam organizados de forma estruturada e com interconexões. Elmasri e Navathe, em 2011, diziam que um banco de dados relacional permite armazenar, consultar e manipular grandes quantidades de dados, mantendo sua integridade e coerência.

Além disso, o modelo relacional, baseado nos princípios matemáticos da álgebra relacional, oferece vantagens significativas como a redução de redundância, a preservação da integridade dos dados, e ainda, facilita a criação de relatórios. Toda tabela possui uma chave primária, garantindo a unicidade dos registros; as vezes há chaves estrangeiras que demonstram as ligações entre as tabelas.

### **2.2 Modelagem de Dados e Normalização**

A modelagem de dados, fundamental no desenvolvimento de sistemas info, e muito importante. Ela representa informações e suas ligações, com dados bem estruturados e lógicos. Date (2000) ressalta como a modelagem ajuda a entender e organizar as consultas de dados, dentru duma organização.

O Modelo Entidade-Relacionamento (MER), criado por Peter Chen em 1976, usado bastante na modelagem conceitual de bancos de dados. Ele mostra entidades, seus atributos e relacionamentos, servindo de base pro desenvolvimento de modelos lógicos e físicos, coisa crucial!

Além da modelagem, a normalização, um processo chave que elimina redundâncias e garante a integridade dos dados. Segundo Heuser (2009), ela estrutura os dados em formas normais, a maioria dos bancos usando até a terceira forma normal (3FN), pra balancear eficiência e desempenho, algo bem importante.



## **2.3 Trabalhos Relacionados**

Muitos sistemas pra gestão de academias surgiram por aí, como o Gympass, Techfit e GymMaster, que prometem ser completos, cuidando de tudo, desde matrículas, até pagamentos, treinos, e relatórios. Só que, esses programas, na maioria das vezes, são caros, e você não pode mexer neles como quiser, nem ver seus dados. O que dificulta, né, as adaptações.

Por outro lado, boa parte desses sistemas funcionam mais online, só que este projeto quer criar um banco de dados, tipo, dentro do computador mesmo, um sistema local, pra atender as necessidades de uma academia menor ou média, e com chances de crescer, no futuro.

Esses foram os pilares teóricos que guiaram a criação do Sistema de Gestão de Academia, que a gente tá construindo aqui. Na parte seguinte, vamos mostrar como vai ser o projeto, começando com o que a academia precisa, passando pelo desenho dos dados e até chegar no banco de dados pronto.

### 3 PROJETO PROPOSTO

Nesta seção, a gente mergulha nos estágios de desenvolvimento do Sistema de Gestão da Academia, cobrindo a coleta de requisitos e a modelagem do banco de dados.

Também descrevemos as entidades, suas ligações complexas e os artefatos criados, tipo o Diagrama Entidade-Relacionamento, que, a gente chama de "pé de galinha".

O objetivo principal deste projeto é garantir que toda informação essencial para a gestão da academia seja bem organizada, mantendo a segurança e a precisão dos dados.

#### 3.1 Apresentação de Figuras

O Modelo Conceitual, representado pelo Diagrama Entidade-Relacionamento (DER), é uma das principais ferramentas para o desenvolvimento de um banco de dados relacional. Este diagrama representa de forma visual as entidades, seus

atributos e os relacionamentos existentes entre elas.

Table	
PK	<u>ID_Aluno</u>
	Nome
	CPF
	Endereço
	Telefone
	Email
	Data_Nascimento
FK	ID_Plano

Table	
PK	<u>ID_Plano</u>
	Descrição
	Valor
	Duração

Table	
PK	<u>ID_Frequência</u>
	Data
	Hora_Entrada
FK	ID_Aluno

Table	
PK	<u>ID_Pagamento</u>
	Data
	Valor
	Forma_Pagamento
FK	ID_Aluno

Table	
PK	<u>ID_Treino</u>
	Data_Criação
	Observações
FK	ID_Aluno

Table	
PK	<u>ID_Item_Treino</u>
	Séries
	Repetições
	Carga
FK	ID_Treino
FK	ID-Exercício

### 3.2 Descrição das Entidades

Aluno: Responsável por armazenar os dados dos alunos da academia, como nome, CPF, endereço, telefone e e-mail. Cada aluno é identificado por um código único (ID\_Aluno).

- Plano: Contém informações sobre os planos de assinatura da academia, incluindo descrição, valor e duração. Cada plano possui um identificador único (ID\_Plano).
- Pagamento: Registra os pagamentos realizados pelos alunos, incluindo data, valor, forma de pagamento e vínculo ao aluno. Essa entidade permite o controle financeiro da academia.
- Frequência: Armazena os registros de presença dos alunos, com data e hora de entrada. Permite acompanhar a assiduidade dos alunos na academia.
- Treino: Contém as fichas de treino de cada aluno, incluindo data de criação, observações e responsável técnico.
- Exercício: Lista os exercícios disponíveis na academia, com informações como nome, grupo muscular, descrição e orientações.
- Itens\_Treino: Entidade associativa que relaciona os treinos aos exercícios, indicando quantas séries, repetições e cargas são recomendadas para cada exercício no treino de um aluno.

### 3.3 Descrição dos Relacionamentos

- **Aluno - Plano:** Cada aluno está vinculado a um plano. O relacionamento é de muitos para um (N:1), onde muitos alunos podem estar associados a um mesmo plano.
- **Aluno - Pagamento:** Um aluno pode ter vários pagamentos ao longo do tempo. Relacionamento de um para muitos (1:N).
- **Aluno - Frequência:** Cada vez que o aluno comparece à academia, é registrado na tabela de frequência. Relacionamento de um para muitos (1:N).

- **Aluno - Treino:** Um aluno pode ter uma ou mais fichas de treino. Relacionamento de um para muitos (1:N).
- **Treino - Itens\_Treino:** Cada ficha de treino possui vários itens que representam os exercícios realizados. Relacionamento de um para muitos (1:N).
- **Itens\_Treino - Exercício:** Cada item de treino está associado a um exercício específico. Relacionamento de muitos para um (N:1).

### 3.4 Apresentação de Algoritmos

#### Algoritmo 1 – Criação da Tabela Aluno

```
1 CREATE TABLE Aluno (  
2     ID_Aluno INT PRIMARY KEY AUTO_INCREMENT,  
3     Nome VARCHAR(100) NOT NULL,  
4     CPF VARCHAR(14) NOT NULL UNIQUE,  
5     Endereco VARCHAR(200),  
6     Telefone VARCHAR(15),  
7     Email VARCHAR(100),  
8     Data_Nascimento DATE,  
9     ID_Plano INT,  
10    FOREIGN KEY (ID_Plano) REFERENCES Plano(ID_Plano)]  
11 );  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

**Algoritmo 2 – Criação da Tabela Plano**

```
1 CREATE TABLE Plano (  
2     ID_Plano INT PRIMARY KEY AUTO_INCREMENT,  
3     Descricao VARCHAR(100) NOT NULL,  
4     Valor DECIMAL(10,2) NOT NULL,  
5     Duracao INT NOT NULL  
6 );  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

### Algoritmo 3 – Criação da Tabela Pagamento

```
1 CREATE TABLE Pagamento (  
2     ID_Pagamento INT PRIMARY KEY AUTO_INCREMENT,  
3     Data DATE NOT NULL,  
4     Valor DECIMAL(10,2) NOT NULL,  
5     Forma_Pagamento VARCHAR(50),  
6     ID_Aluno INT,  
7     FOREIGN KEY (ID_Aluno) REFERENCES Aluno(ID_Aluno)  
8 );  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

### Algoritmo 4 – Criação da Tabela Frequência

```
1 CREATE TABLE Frequencia (  
2     ID_Frequencia INT PRIMARY KEY AUTO_INCREMENT,  
3     Data DATE NOT NULL,  
4     Hora_Entrada TIME,  
5     ID_Aluno INT,  
6     FOREIGN KEY (ID_Aluno) REFERENCES Aluno(ID_Aluno)  
7 );  
8  
9  
10
```

11  
12  
13  
14  
15  
16

### **Algoritmo 5 – Criação da Tabela Exercício**

```
1 CREATE TABLE Exercicio (  
2     ID_Exercicio INT PRIMARY KEY AUTO_INCREMENT,  
3     Nome VARCHAR(100) NOT NULL,  
4     Grupo_Muscular VARCHAR(50),  
5     Descricao VARCHAR(200)  
6 );  
7  
8
```



**Algoritmo 6 – Criação da Tabela Treino**

```
1 CREATE TABLE Treino (  
2     ID_Treino INT PRIMARY KEY AUTO_INCREMENT,  
3     Data_Criacao DATE NOT NULL,  
4     Observacao VARCHAR(200),  
5     ID_Aluno INT,  
6     FOREIGN KEY (ID_Aluno) REFERENCES Aluno(ID_Aluno)  
7 );  
8  
9
```

**Algoritmo 7 – Criação da Tabela Itens\_Treino**

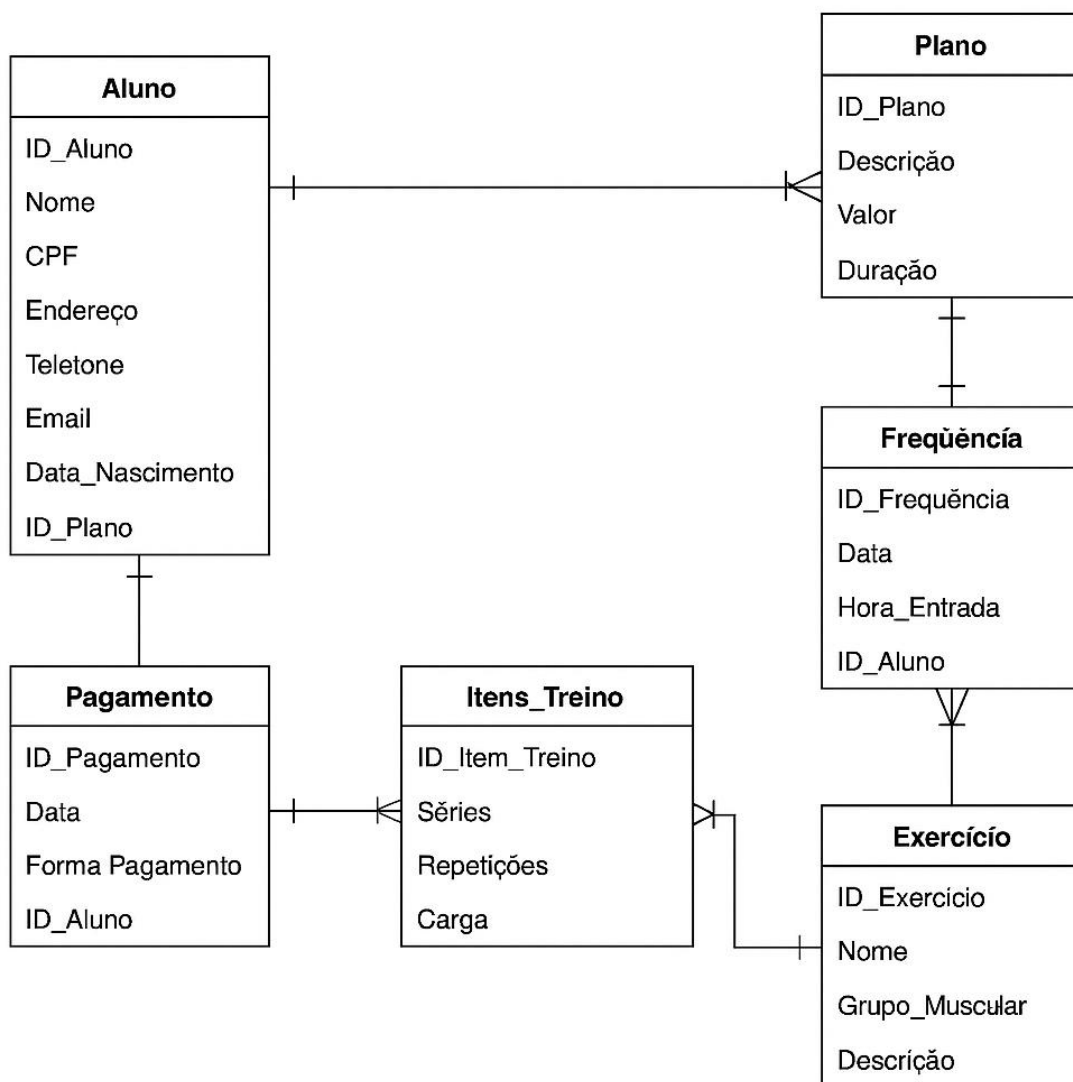
```
1 CREATE TABLE Itens_Treino (  
2     ID_Item_Treino INT PRIMARY KEY AUTO_INCREMENT,  
3     Series INT NOT NULL,  
4     Repeticoes INT NOT NULL,  
5     Carga DECIMAL(5,2),  
6     ID_Treino INT,  
7     ID_Exercicio INT,  
8     FOREIGN KEY (ID_Treino) REFERENCES Treino(ID_Treino),  
9     FOREIGN KEY (ID_Exercicio) REFERENCES Exercicio(ID_Exercicio)  
10 );
```

### 3.5 Resultados

Nesta seção são apresentados os resultados obtidos a partir do desenvolvimento do banco de dados relacional para o **Sistema de Gestão de Academia**. Foram desenvolvidos os modelos conceitual, lógico e físico, bem como o dicionário de dados, regras de negócio implementadas e testes realizados através de consultas SQL, que comprovam o correto funcionamento da estrutura proposta.

#### 3.5.1 Modelo Conceitual

O modelo conceitual foi elaborado utilizando a notação Entidade-Relacionamento (DER) no formato pé de galinha. Ele apresenta sete entidades principais: **Aluno**, **Plano**, **Pagamento**, **Frequência**, **Treino**, **Exercício** e **Itens\_Treino**, além de seus respectivos relacionamentos e restrições.



### Regras de Negócio

- Cada aluno está vinculado a um único plano.
- Um aluno pode ter vários pagamentos.
- Um aluno pode ter várias frequências registradas.
- Um aluno pode ter vários treinos.
- Cada treino possui vários exercícios vinculados via Itens\_Treino.
- Cada exercício pode pertencer a vários treinos diferentes.

### 3.5.2 Dicionário de Dados

Entidade	Descrição
<b>Aluno</b>	Cadastro dos alunos
<b>Plano</b>	Dados dos planos da academia
<b>Pagamento</b>	Controle dos pagamentos
<b>Frequência</b>	Controle de entrada dos alunos
<b>Treino</b>	Cadastro de ficha de treino dos alunos
<b>Exercício</b>	Lista de exercícios disponíveis
<b>Itens_Treino</b>	Ligação entre treino e exercício com dados de séries, repetições e carga

### 3.5.3 Consultas SQL Desenvolvidas

Nº	Descrição da Consulta	Código SQL
1	Listar alunos e seus planos	SELECT A.Nome, P.Descricao AS Plano FROM Aluno A JOIN Plano P ON A.ID_Plano = P.ID_Plano;
2	Pagamentos acima de R\$200	SELECT ID_Pagamento, ID_Aluno, Valor, Data FROM Pagamento WHERE Valor > 200;
3	Alunos com mais de 3 frequências	SELECT ID_Aluno, COUNT(*) AS Total_Frequencias FROM Frequencia GROUP BY ID_Aluno HAVING Total_Frequencias > 3;
4	Soma dos pagamentos por aluno	SELECT ID_Aluno, SUM(Valor) AS Total_Pago FROM Pagamento GROUP BY ID_Aluno;
5	Exercícios do grupo 'Perna'	SELECT Nome, Grupo_Muscular FROM Exercicio WHERE Grupo_Muscular = 'Perna';
6	Alunos sem treino cadastrado	SELECT A.Nome FROM Aluno A LEFT JOIN Treino T ON A.ID_Aluno = T.ID_Aluno WHERE T.ID_Treino IS NULL;
7	Quantidade de treinos por	SELECT A.Nome,

	aluno	COUNT(T.ID_Treino) AS Total_Treinos FROM Aluno A JOIN Treino T ON A.ID_Aluno = T.ID_Aluno GROUP BY A.Nome;
8	Última frequência de cada aluno	SELECT ID_Aluno, MAX(Data) AS Ulti- ma_Frequencia FROM Frequencia GROUP BY ID_Aluno;
9	Último pagamento de cada aluno	SELECT ID_Aluno, MAX(Data) AS Ulti- mo_Pagamento FROM Pagamento GROUP BY ID_Aluno;
10	Média dos valores de pagamentos	SELECT AVG(Valor) AS Media_Pagamento FROM Pagamento;
11	Alunos e seus últimos pagamentos	SELECT A.Nome, MAX(P.Data) AS Ulti- mo_Pagamento FROM Aluno A JOIN Pagamento P ON A.ID_Aluno = P.ID_Aluno GROUP BY A.Nome;
12	Treinos e seus exercícios	SELECT T.ID_Treino, E.Nome AS Exercicio FROM Itens_Treino IT JOIN Exercicio E ON IT.ID_Exercicio = E.ID_Exercicio JOIN Treino T ON IT.ID_Treino = T.ID_Treino;
13	Total de alunos por plano	SELECT ID_Plano, COUNT(*) AS To- tal_Alunos FROM Aluno GROUP BY ID_Plano;
14	Alunos que pagaram em janeiro de 2025	SELECT ID_Aluno FROM Pagamento WHERE MONTH(Data) = 1 AND YEAR(Data) = 2025;
15	Soma dos pagamentos por plano	SELECT A.ID_Plano, SUM(P.Valor) AS To- tal_Pago FROM Aluno A JOIN Pagamento P ON A.ID_Aluno = P.ID_Aluno GROUP BY A.ID_Plano;
16	Alunos sem pagamentos	SELECT A.Nome FROM Aluno A LEFT JOIN Pagamento P

17	Frequência média dos alunos	<pre> ON A.ID_Aluno = P.ID_Aluno WHERE P.ID_Pagamento IS NULL; SELECT AVG(Freq.Total_Frequenci as) AS Me- dia_Frequencias FROM (SELECT ID_Aluno, COUNT(*) AS Total_Frequencias FROM Frequencia GROUP BY ID_Aluno) AS Freq; </pre>
18	Alunos e total de exercícios nos treinos	<pre> SELECT A.Nome, COUNT(IT.ID_Item_Treino ) AS Total_Exercicios FROM Aluno A JOIN Treino T ON A.ID_Aluno = T.ID_Aluno JOIN Itens_Treino IT ON T.ID_Treino = IT.ID_Treino GROUP BY A.Nome; </pre>
19	Top 3 alunos que mais frequentaram	<pre> SELECT ID_Aluno, COUNT(*) AS To- tal_Frequencias FROM Frequencia GROUP BY ID_Aluno ORDER BY To- tal_Frequencias DESC LIMIT 3; </pre>
20	Top 5 alunos com maior total de pagamentos	<pre> SELECT ID_Aluno, SUM(Valor) AS To- tal_Pago FROM Pagamento GROUP BY ID_Aluno ORDER BY Total_Pago DESC LIMIT 5; </pre>

## 4 CONCLUSÃO

O desenvolvimento do projeto Sistema de Gestão de Academia permitiu alcançar plenamente os objetivos propostos. Através da construção do banco de dados relacional, foi possível organizar e estruturar as informações necessárias para o gerenciamento eficiente de uma academia, tais como controle de alunos, planos, pagamentos, treinos e frequência.

Cada etapa do desenvolvimento foi fundamental para garantir a consistência dos dados e a integridade das informações. O modelo conceitual, representado pelo Diagrama Entidade-Relacionamento (DER), permitiu mapear corretamente as entidades e seus relacionamentos, refletindo fielmente as regras de negócio identificadas. O modelo lógico transformou esses conceitos em tabelas relacionais com seus respectivos atributos, chaves primárias e estrangeiras. Por fim, o modelo físico concretizou a estrutura no SGBD MySQL, permitindo a realização de testes, inserções e consultas eficientes.

As 20 consultas SQL elaboradas comprovaram a eficácia do banco de dados, permitindo análises detalhadas sobre o funcionamento da academia, como controle financeiro, acompanhamento da assiduidade dos alunos, distribuição dos treinos e gestão dos planos.

Com relação ao cronograma previamente estabelecido, todas as atividades foram cumpridas dentro do prazo, desde o levantamento de requisitos até a implementação final do banco de dados. Não houve pendências ou atrasos relevantes durante o desenvolvimento deste projeto.

Portanto, conclui-se que os objetivos foram plenamente alcançados, e o sistema proposto atende às demandas de gestão de academias de pequeno e médio porte, oferecendo uma solução eficiente, segura e de fácil manutenção. Além disso, este trabalho proporcionou um valioso aprendizado prático sobre modelagem de dados, normalização, SQL e desenvolvimento de sistemas de banco de dados.



## REFERÊNCIAS

ELMASRI, Ramez; NAVATHE, Shamkant. Sistemas de Banco de Dados. 6. ed. São Paulo: Pearson, 2011.

- HEUSER, Carlos Alberto. Projeto de Banco de Dados. 6. ed. Porto Alegre: Bookman, 2009.
- DATE, C. J. Introdução a Sistemas de Bancos de Dados. 8. ed. Rio de Janeiro: Elsevier, 2004.
- CHEN, Peter. The Entity-Relationship Model—Toward a Unified View of Data. ACM Transactions on Database Systems, v. 1, n. 1, p. 9-36, 1976.
- BUSINESS-INTELLIGENCE. Performance measurement is now a critical issue for the HR professional. Disponível em: <http://www.business-intelligence.co.uk/reports/hrscorecard/>.
- MYSQL. MySQL 8.0 Reference Manual. Disponível em: <https://dev.mysql.com/doc/>.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. NBR 6023: informação e documentação — Referências — Elaboração. Rio de Janeiro, 2018.