



**CENTRO UNIVERSITÁRIO BRASILEIRO - UNIBRA**  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE SISTEMA DE GESTÃO PARA  
ACADEMIAS: REQUISITOS, MODELAGEM E EXECUÇÃO**

RECIFE

2024

**CENTRO UNIVERSITÁRIO BRASILEIRO - UNIBRA**  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE SISTEMA DE GESTÃO PARA  
ACADEMIAS: REQUISITOS, MODELAGEM E EXECUÇÃO**

Projeto interdisciplinar apresentado no Centro  
Universitário Brasileiro - **UNIBRA**, Curso  
Análise e Desenvolvimento de Sistemas  
<**ADS 2NA**>, como pré-requisito avaliativo  
para a obtenção parcial da 2º avaliação do  
semestre.

**Fomentador:** Profº. Bruno Roberto Silva.

## RESUMO

Este trabalho apresenta os requisitos para o desenvolvimento de um sistema de academia, visando facilitar o agendamento das aulas, melhorar o gerenciamento das atividades e proporcionar uma experiência satisfatória aos usuários. Quanto aos requisitos funcionais, destaca-se a necessidade de cadastro de usuário, login, redefinição de senha, quadro de Horários de Aula, controle de frequência, cadastro na Confederação Brasileira de Jiu Jitsu (CBJJ), cadastro de funcionários, atribuição de horários aos professores. Quanto aos requisitos não funcionais, são abordados aspectos como usabilidade, segurança, confiabilidade, desempenho, manutenção e portabilidade. O desenvolvimento do sistema inclui a criação de diagrama de classes e diagrama de casos de uso. A modelagem de banco de dados envolve a criação de diagramas Entidade-Relacionamento (ER), modelo relacional e esquema físico. A implementação é realizada com tecnologias como MySQL, SQL, Windows, WAMP Server, BrModelo, MySQL Workbench, lucidchart e trello.

**Palavras-chave:** academia, sistema, requisitos, funcionalidades, desenvolvimento, banco de dados, implementação, validação, tecnologias, segurança.

## **ABSTRACT**

This work presents the requirements for the development of a gym system, aiming to facilitate class scheduling, improve activity management and provide a satisfactory experience for users. Regarding functional requirements, the need for user registration, login, password reset, Class Schedule table, frequency control, registration with the Brazilian Jiu Jitsu Confederation (CBJJ), employee registration, assignment of schedules to students stands out. teachers. As for non-functional requirements, aspects such as usability, security, reliability, performance, maintenance and portability are addressed. System development includes creating class diagram and use case diagram. Database modeling involves creating Entity-Relationship (ER) diagrams, relational model and physical schema. The implementation is carried out with technologies such as MySQL, SQL, Windows, WAMP Server, BrModelo, MySQL Workbench, lucidchart and trello.

**Keywords:** academia, system, requirements, features, development, database, implementation, validation, technologies, security.

## LISTA DE FIGURAS

Figura 1 - Diagrama de Classe.....	16
Figura 2 - Diagrama Casos de uso.....	17
Figura 3 - Modelo Conceitual.....	20
Figura 4 - Modelo Lógico.....	21
Figura 5 - Modelo Físico.....	22
Figura 6 - Inserção de dados na tabela Usuario.....	22

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>8</b>
1.1 Propósito.....	8
1.2 Objetivo.....	9
1.3 Público Alvo.....	9
<b>2 ENGENHARIA DE REQUISITOS I.....</b>	<b>10</b>
2.1 Requisitos Funcionais.....	10
[RF001] Cadastro de Usuário.....	10
[RF002] Login do Usuário.....	11
[RF003] Redefinição de Senha.....	11
[RF004] Quadro de Horários de Aula.....	11
[RF005] Sistema de Presença.....	11
[RF006] Registro do CBJJ.....	12
[RF007] Cadastro de Funcionários.....	12
[RF008] Controle de Presença dos Professores.....	12
[RF009] Suporte.....	12
2.2 Requisitos Não-funcionais.....	13
[RNF001] Usabilidade.....	13
[RNF002] Segurança.....	13
[RNF003] Confiabilidade.....	13
[RNF004] Desempenho.....	13
[RNF005] Manutenção.....	14
[RNF006] Portabilidade.....	14
2.3 Resumo do Capítulo.....	14
<b>3 LINGUAGEM DE PROGRAMAÇÃO ORIENTADA À OBJETOS.....</b>	<b>15</b>
3.1 Diagrama de classe.....	16
3.2 Diagrama de caso de uso.....	17
3.3 Resumo do Capítulo.....	18
<b>4 PROGRAMAÇÃO E OTIMIZAÇÃO DE BANCO DE DADOS.....</b>	<b>19</b>
4.1 Diagrama de Entidade-Relacionamento (ER) .....	20
4.2 Diagrama de Modelo Relacional.....	21
4.3 Esquema de banco de dados físico.....	22
4.4 Resumo do Capítulo.....	23
<b>5 REDES E AMBIENTES OPERACIONAIS II.....</b>	<b>24</b>
5.1 Resumo do Capítulo.....	25
<b>6 ANÁLISE DO IMPACTO NO NEGÓCIO.....</b>	<b>26</b>
6.1 Sucesso do Projeto.....	26
6.2 Feedback da Empresa.....	26
6.3 Dificuldades Enfrentadas.....	27
<b>7 CONCLUSÃO.....</b>	<b>28</b>

**REFERÊNCIAS.....29**  
**APÊNDICE A - EQUIPE.....30**

## **1 INTRODUÇÃO**

Este trabalho explora a implementação de um site inovador na Academia Perfil Cross, visando adaptar-se às necessidades de uma sociedade cada vez mais conectada e digitalizada. Em um mundo onde a tecnologia desempenha um papel central, academias como a Perfil Cross buscam soluções eficazes para aprimorar a experiência do cliente e otimizar suas operações.

O site proposto apresenta uma variedade de funcionalidades, incluindo quadro de horários de aula, interação direta com instrutores e, futuramente, acesso a conteúdos exclusivos (Vídeos de Treinamento, Planos de Treinamento Personalizados, Artigos e Recursos de Saúde e Bem-Estar, Desafios e Competições). Ao oferecer tais recursos, o objetivo é melhorar significativamente a experiência dos clientes da Academia Perfil Cross, ao mesmo tempo em que otimiza a gestão interna."

Este trabalho oferece insights valiosos para gestores, profissionais do fitness e demais interessados no futuro da academia. Além disso, discute-se os impactos esperados dessa inovação tecnológica na gestão da Academia Perfil Cross, na experiência do cliente e nas estratégias de crescimento do negócio.

Por meio de uma análise aprofundada, busca-se não apenas destacar as potencialidades deste site inovador, mas também inspirar a adoção de práticas que impulsionem o progresso e a competitividade da Academia Perfil Cross neste contexto em constante evolução.

### **1.1 Propósito**

Especifica os requisitos do sistema de academia a ser desenvolvido, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema. Foi cuidadosamente desenvolvido para oferecer acesso fácil e conveniente aos serviços e informações essenciais relacionados à academia.



## 1.2 Objetivo

Facilitar o processo de agendamento de aulas para os alunos da academia, Melhorar a gestão das aulas pela academia, Proporcionar uma melhor experiência para os alunos.

## 1.3 Público Alvo

**Usuários finais:** Alunos da academia que desejam utilizar o sistema para agendar aulas, registrar presença e outras funcionalidades.

**Equipe técnica:** Desenvolvedores, administradores de sistemas e outros profissionais que precisam entender a arquitetura, implementação e funcionamento do sistema.

**Stakeholders:** Proprietários da academia, gerentes, investidores e outros interessados no projeto.

## **2 ENGENHARIA DE REQUISITOS I**

Este capítulo visa fornecer uma análise completa dos requisitos funcionais e não funcionais do sistema web para uma academia de artes marciais, com o objetivo de identificar as necessidades e preferências dos usuários, estabelecendo assim uma base clara para o desenvolvimento do projeto.

Para o desenvolvimento do projeto de engenharia de requisitos para a academia de artes marciais, foram utilizadas diversas tecnologias e ferramentas para garantir uma gestão eficiente e uma análise completa dos requisitos funcionais e não funcionais do sistema web.

Para a diagramação e visualização dos requisitos, foi adotada a ferramenta Lucidchart. Essa plataforma oferece uma interface intuitiva e recursos robustos para criar diagramas claros e detalhados, facilitando a compreensão e comunicação dos requisitos entre os membros da equipe de desenvolvimento e os stakeholders.

Além disso, para a gestão do projeto e acompanhamento das tarefas, foi empregado o Trello. Essa ferramenta de gestão de projetos baseada em quadros Kanban permite organizar as atividades em cartões, atribuir responsabilidades, definir prazos e acompanhar o progresso de forma visual e colaborativa. Com o Trello, a equipe pôde manter-se organizada e focada nas entregas, garantindo o cumprimento dos requisitos dentro do cronograma estabelecido.

Dessa forma, a combinação entre Lucidchart e Trello proporcionou uma abordagem integrada e eficaz para a engenharia de requisitos, desde a análise inicial até a gestão do projeto, assegurando que o sistema web para a academia de artes marciais atenda plenamente às necessidades e expectativas dos usuários e stakeholders.

### **2.1 Requisitos Funcionais**

#### **[RF001] Cadastro de Usuário**

O site deve permitir o cadastro de um novo usuário com as informações mínimas como: nome completo, telefone, e-mail, senha e data de nascimento.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### **[RF002] Login do Usuário**

O site deve permitir que o usuário faça login utilizando seu e-mail e senha.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### **[RF003] Redefinição de Senha**

O site deve permitir que um usuário redefina sua senha caso a esqueça. Uma mensagem será enviada para o e-mail cadastrado na conta, fornecendo instruções para a redefinição da senha.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### **[RF004] Quadro de Horários de Aula**

Deve haver um quadro de horários de aula disponível para as modalidades de Muay Thai, Jiu Jitsu e CrossFit.

Os horários das aulas devem ser atualizados regularmente e acessíveis aos usuários.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### **[RF005] Sistema de Presença**

O site deve permitir o registro de presença dos alunos nas aulas agendadas. Deve haver uma forma de monitorar a presença dos alunos de forma eficiente e precisa.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

**[RF006] Registro do CBJJ**

Para os alunos de Jiu Jitsu, o sistema deve possibilitar o registro junto à Confederação Brasileira de Jiu Jitsu (CBJJ), se aplicável.

**Prioridade:**     ☒ Essencial                      ☐ Importante                      ☐ Desejável

**[RF007] Cadastro de Funcionários**

O site deve permitir o cadastro dos funcionários, especificamente os professores de Muay Thai, Jiu Jitsu e CrossFit.

Deve ser possível registrar informações como nome, telefone de contato e especialidade (modalidade).

**Prioridade:**     ☒ Essencial                      ☐ Importante                      ☐ Desejável

**[RF008] Controle de Presença dos Professores**

O site deve permitir o registro de presença dos professores nas aulas que lecionam.

Deve ser possível monitorar a presença dos professores para garantir a continuidade das atividades.

**Prioridade:**     ☒ Essencial                      ☐ Importante                      ☐ Desejável

**[RF009] Suporte**

O site terá um suporte de forma integral para resolver problemas e tirar dúvidas do usuário. Respondendo em até 24 horas.

**Prioridade:**     ☒ Essencial                      ☐ Importante                      ☐ Desejável

## 2.2 Requisitos Não-funcionais

### [RNF001] Usabilidade

O sistema deve ser intuitivo e fácil de usar, mesmo para usuários iniciantes. Deve ser responsivo e acessível em diferentes dispositivos, como computadores, tablets e smartphones.

O usuário deve ser capaz de realizar qualquer funcionalidade da aplicação em no máximo 5 minutos.

**Prioridade:**      ☐ Essencial      ☒ Importante      ☐ Desejável

### [RNF002] Segurança

Deve garantir a segurança das informações dos usuários, utilizando métodos adequados de proteção de dados.

**Prioridade:**      ☒ Essencial      ☐ Importante      ☐ Desejável

### [RNF003] Confiabilidade

Deve ser realizado um backup regular dos dados para evitar perda de informações importantes.

**Prioridade:**      ☒ Essencial      ☐ Importante      ☐ Desejável

### [RNF004] Desempenho

O sistema deve ser capaz de lidar com um grande número de usuários simultâneos, garantindo tempos de resposta rápidos.

As atualizações do quadro de horários e registros de presença devem ser processadas de forma eficiente e em tempo real.

**Prioridade:**      ☒ Essencial      ☐ Importante      ☐ Desejável

**[RNF005] Manutenção**

Deve ser fácil de realizar manutenção e atualizações no sistema, garantindo sua escalabilidade e adaptabilidade a novas necessidades.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

**[RNF006] Portabilidade**

O sistema deve ser executado em computadores com 1.8GHz ou superior, 1GB de ram ou superior, com sistema operacional Windows 7 ou acima.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

**2.3 Resumo do Capítulo**

Apresenta os requisitos funcionais e não funcionais do sistema web para academia de artes marciais, detalhando as funcionalidades, características e restrições técnicas do sistema. As informações aqui descritas servem como base para o desenvolvimento do projeto, garantindo que o sistema atenda às necessidades dos usuários e stakeholders.

### 3 LINGUAGEM DE PROGRAMAÇÃO ORIENTADA À OBJETOS

O capítulo sobre Linguagem de Programação Orientada a Objetos neste projeto aborda a etapa do desenvolvimento do sistema de academia que compreende a modelagem e implementação do código, empregando os princípios da orientação a objetos e utilizando Java com Spring Boot.

Para a implementação do sistema de academia, foram empregadas diversas tecnologias e ferramentas tanto para o desenvolvimento do front-end quanto do back-end. No âmbito do front-end, foram utilizadas tecnologias fundamentais para a criação da interface do usuário (UI) e a interatividade do sistema. O HTML (Hypertext Markup Language) foi empregado para estruturar o conteúdo das páginas web, o CSS (Cascading Style Sheets) para estilizar e formatar essas páginas, e o JavaScript (JS) para adicionar interatividade e dinamismo à aplicação, possibilitando a execução de ações e o processamento de dados diretamente no navegador do usuário.

Já no back-end, a linguagem de programação escolhida foi o Java, conhecida pela sua robustez, portabilidade e ampla adoção na indústria de desenvolvimento de software. Complementarmente, o framework Spring Boot foi utilizado para simplificar e acelerar o desenvolvimento de aplicativos Java, fornecendo um ambiente de configuração mínimo e recursos prontos para uso, como injeção de dependência, segurança e acesso a dados.

Essas tecnologias e ferramentas foram escolhidas com base em sua capacidade de fornecer uma solução sólida e escalável para o desenvolvimento do sistema de academia, garantindo uma experiência de usuário intuitiva e eficiente, além de facilitar a manutenção e evolução da aplicação ao longo do tempo.

O desenvolvimento do sistema de academia envolveu uma abordagem abrangente que contemplou tanto a interface do usuário quanto a lógica de negócios por trás dela. No front-end, a combinação de HTML, CSS e JavaScript permitiu criar uma experiência de usuário rica e dinâmica, possibilitando a interação fluida com o sistema diretamente no navegador (W3SCHOOLS, 2024).

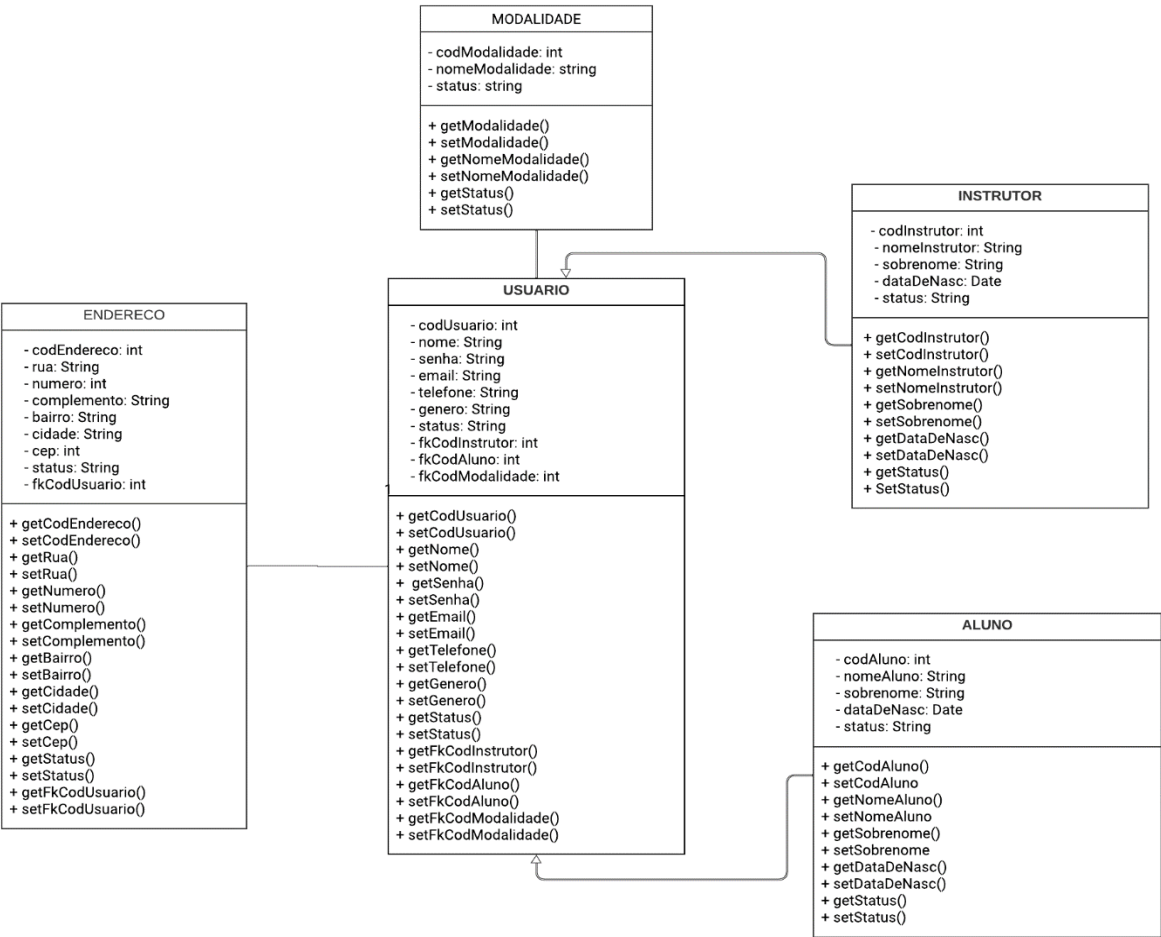
Por outro lado, no back-end, a escolha do Java aliado ao Spring Boot proporcionou uma base sólida para a implementação da lógica de negócios, garantindo não apenas robustez e portabilidade, mas também facilitando a manutenção e o desenvolvimento contínuo do sistema. Essa integração harmoniosa entre front-end e back-end resultou

em um sistema de academia completo e eficiente, capaz de atender às necessidades dos usuários de forma intuitiva e escalável, mantendo-se alinhado com as melhores práticas de desenvolvimento de software (MAVEN, 2024).

3.1 Diagrama de classe

Classe é um modelo ou blueprint para criar objetos que compartilham características semelhantes. Na imagem apresentada, vemos que o usuário é a base, contendo um nome de usuário para login e um endereço. Um instrutor, uma subclasse de usuário, além desses atributos, também leciona uma modalidade. Similarmente, um aluno, outra subclasse de usuário. O endereço é uma entidade separada com atributos como rua, cidade, estado e CEP. Essa estrutura permite a criação de objetos que compartilham características semelhantes, organizados hierarquicamente.

Figura 1 - Diagrama de Classe



Fonte: Autoria própria (2024).



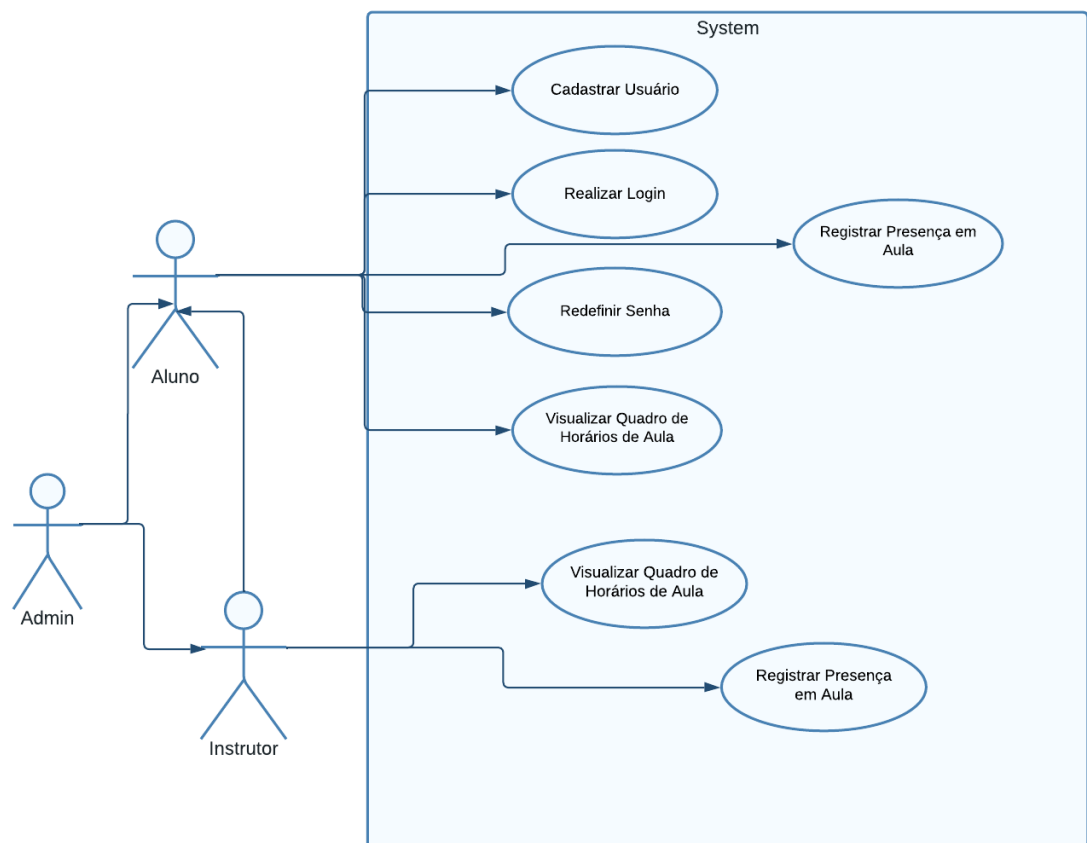
### 3.2 Diagrama de caso de uso

Um roteiro detalhado que descreve as interações entre os usuários (atores) e as funcionalidades do sistema (casos de uso).

A imagem fornecida apresenta um exemplo de UML. Nele, são identificados os atores "Aluno", "Instrutor" e "Admin".

Os **alunos** se cadastram e acessam o sistema para realizar diversas tarefas. Os **instrutores** gerenciam as aulas. O **administrador** é responsável por gerenciar o sistema e os usuários.

Figura 2 - Diagrama Casos de uso



Fonte: Autoria própria (2024).

### **3.3 Resumo do Capítulo**

Apresenta os principais elementos de modelagem e desenvolvimento de sistemas de software, utilizando diferentes ferramentas visuais para representar a estrutura, funcionalidades e fluxo de um sistema.

Inicialmente, é abordado o Diagrama de Classe, que fornece uma representação visual das classes e seus relacionamentos em um sistema orientado a objetos. Em seguida, o Diagrama de Caso de Uso é discutido, destacando os diferentes tipos de usuários e as funcionalidades que podem acessar, servindo como um guia visual das principais interações do sistema.

## 4 PROGRAMAÇÃO E OTIMIZAÇÃO DE BANCO DE DADOS

Para otimizar o desempenho e a eficiência do banco de dados no contexto do sistema de academia, foram empregadas diversas tecnologias e ferramentas especializadas em programação e modelagem de banco de dados.

O banco de dados escolhido para armazenar e gerenciar os dados do sistema foi o MySQL 8.0, uma escolha popular devido à sua confiabilidade, desempenho e recursos avançados de segurança. O MySQL é amplamente utilizado na indústria e oferece suporte para uma variedade de aplicativos e sistemas (MYSQL DOCUMENTATION, 2024).

A linguagem de programação SQL (Structured Query Language) foi utilizada para escrever consultas e comandos que interagem com o banco de dados MySQL. O SQL é uma linguagem padrão para gerenciamento de bancos de dados relacionais e permite realizar operações como inserção, atualização, exclusão e consulta de dados.

Para a modelagem do banco de dados, foi utilizado o ambiente BRModelo. Essa ferramenta oferece recursos para criar diagramas de entidade-relacionamento (ER) e modelos conceituais, permitindo uma representação visual e organizada da estrutura e das relações dos dados (DBEAVER, 2024).

Além disso, o ambiente de desenvolvimento MySQL Workbench foi empregado para realizar tarefas relacionadas à administração, modelagem, monitoramento e desenvolvimento de bancos de dados MySQL. O MySQL Workbench oferece uma interface gráfica intuitiva e recursos poderosos para facilitar o gerenciamento do banco de dados e a criação de consultas SQL.

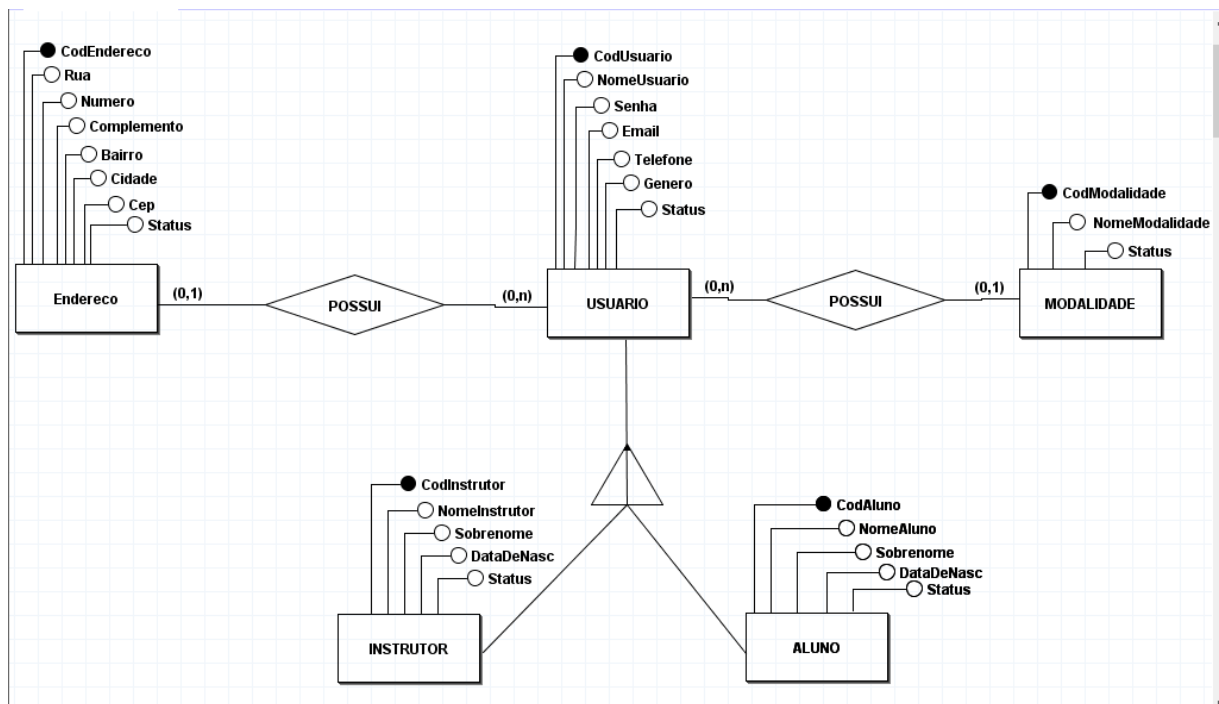
Dessa forma, a combinação das tecnologias e ferramentas mencionadas proporcionou uma abordagem abrangente e eficaz para a programação e otimização do banco de dados do sistema de academia, garantindo uma manipulação eficiente dos dados, consultas rápidas e integridade dos dados.

## 4.1 Diagrama de Entidade-Relacionamento (ER)

O diagrama ajuda a visualizar a estrutura do banco de dados e as relações entre as diferentes entidades. Isso pode ser útil para entender como o sistema funciona e para identificar possíveis problemas.

As entidades estão relacionadas entre si por meio de relacionamentos. Por exemplo, um aluno pode ser treinado por um instrutor.

Figura 3 - Modelo Conceitual



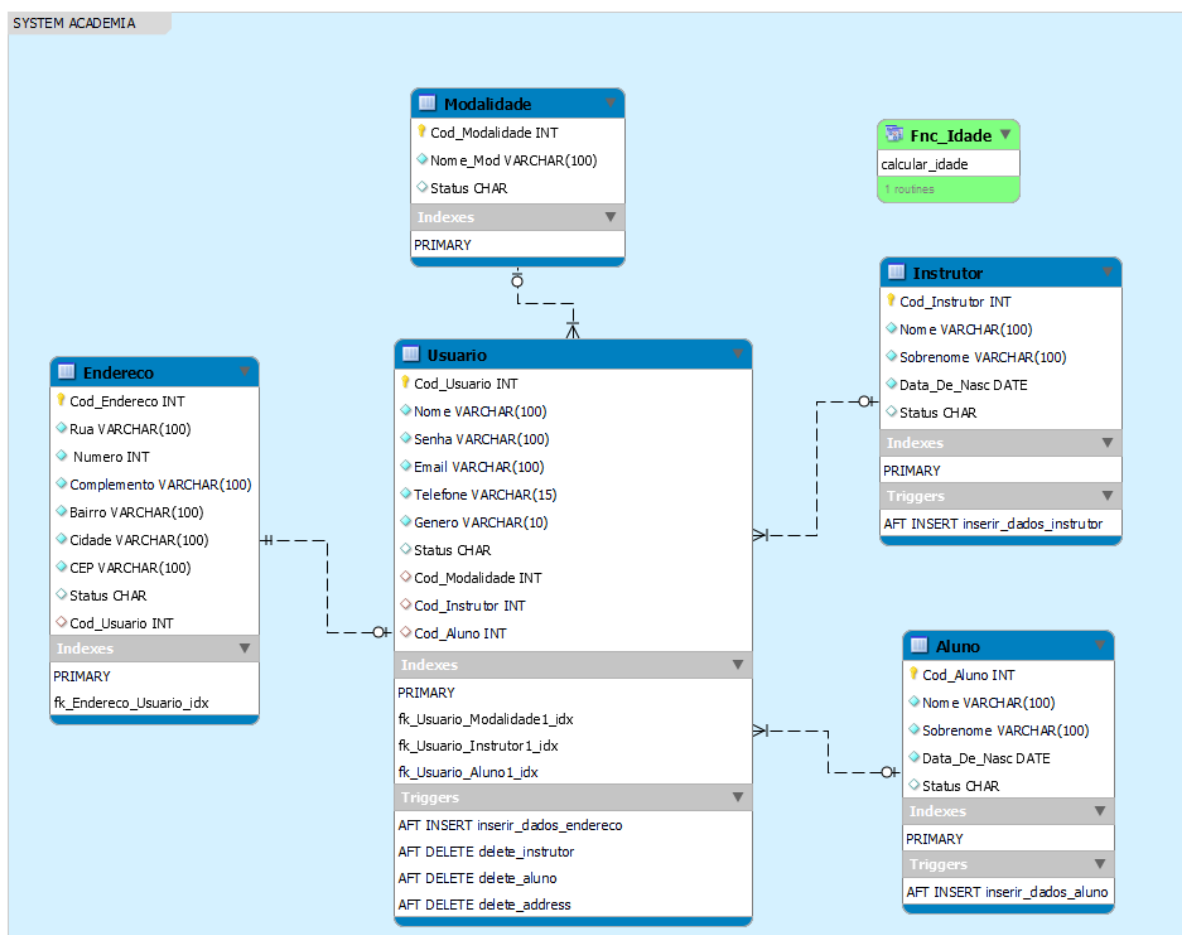
Fonte: Autoria própria (2024).

## 4.2 Diagrama de Modelo Relacional

Representa a estrutura e as relações dos dados de forma abstrata e independente da implementação física, facilitando a comunicação, garantindo a integridade dos dados e fornecendo uma base para o design físico do banco de dados.

- O diagrama utiliza chaves primárias para identificar cada entidade de forma única.
- As chaves estrangeiras são usadas para relacionar as entidades entre si.
- O diagrama segue as regras de normalização para garantir a integridade dos dados.

Figura 4 - Modelo Lógico

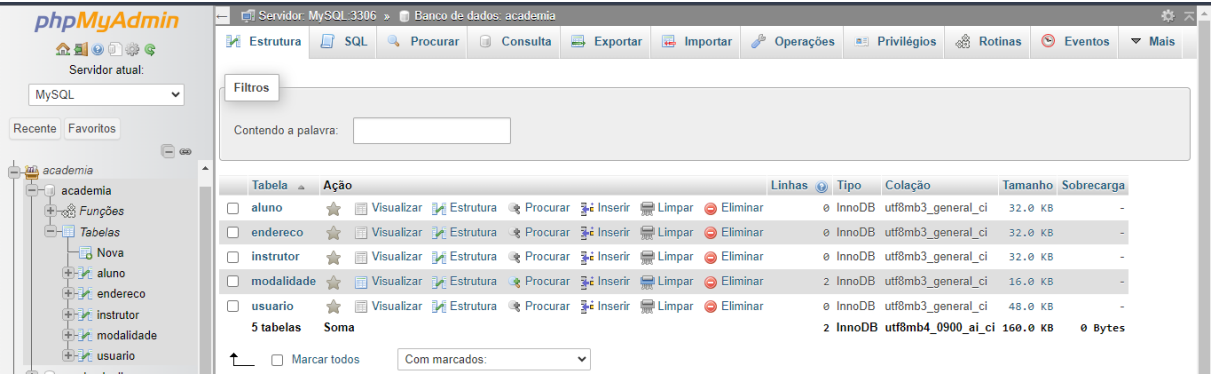


Fonte: Autoria própria (2024).

4.3 Esquema de banco de dados físico

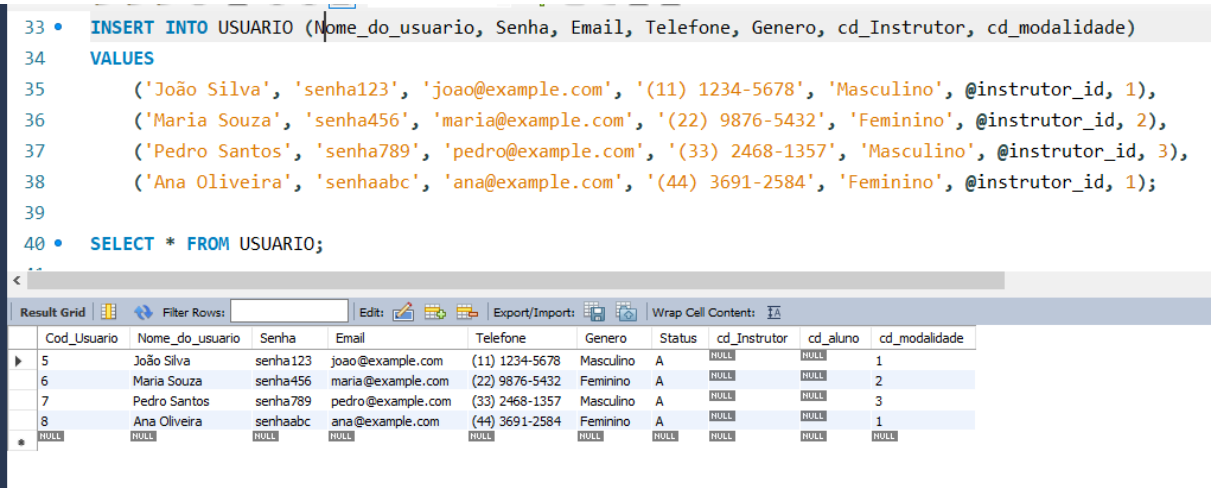
Define como os dados são armazenados em disco, incluindo detalhes como o tipo de armazenamento, índices e otimizações. Isso afeta diretamente o desempenho e a escalabilidade do sistema, garantindo uma manipulação eficiente dos dados e consultas rápidas. Cada tabela no banco de dados tem chaves estrangeiras (FK) que se relacionam com chaves primárias (PK) de outras tabelas, criando vínculos entre os dados. Por exemplo, na tabela "USUARIO", as chaves estrangeiras "Fk\_cd\_instrutor" e "Fk\_cd\_aluno" se relacionam com as chaves primárias das tabelas "INSTRUTOR" e "ALUNO", respectivamente. Isso permite associar um usuário (que pode ser um instrutor ou um aluno) a uma modalidade específica.

Figura 5 - Modelo Físico



Fonte: Autoria própria (2024).

Figura 6 – Inserção de dados na tabela Usuario



Fonte: Autoria própria (2024).

#### 4.4 Resumo do Capítulo

**Modelo Conceitual:** Descreve entidades, atributos e relacionamentos de forma independente de qualquer sistema específico, usando ferramentas como o Modelo Entidade-Relacionamento (MER).

**Modelo Lógico:** Detalha a estrutura das tabelas, chaves e restrições, traduzindo o modelo conceitual para uma linguagem compreensível pelo sistema de gerenciamento de banco de dados (SGBD) escolhido.

**Modelo Físico:** Implementa o banco de dados em um ambiente real, considerando a organização física dos dados, tipos de armazenamento e estratégias de acesso para otimizar o desempenho.

## 5 REDES E AMBIENTES OPERACIONAIS II

A topologia de **Rede Hierárquica** oferece escalabilidade, segurança e flexibilidade. Essa topologia divide a rede em camadas, com dispositivos em cada nível se comunicando com dispositivos do mesmo ou de um nível superior. Isso seria benéfico para a academia, pois permitiria uma organização eficiente da rede, garantindo que diferentes tipos de tráfego possam ser gerenciados de forma separada e que problemas em um nível da rede não afetem diretamente os outros.

Além disso, considerando a natureza do projeto, onde é necessário acesso fácil e seguro aos sistemas e informações relacionados à academia, a utilização de uma rede em nuvem seria altamente recomendada. Os **provedores de nuvem**, como a **Amazon Web Services (AWS)**, oferecem uma plataforma completa e flexível com uma ampla gama de serviços, ideal para aplicações complexas e de grande porte, como o sistema de academia em desenvolvimento. Utilizar a AWS permitiria escalabilidade, alta disponibilidade, segurança avançada e fácil integração com outras ferramentas e serviços necessários para o projeto.

No contexto da implementação de redes e ambientes operacionais para o sistema de academia, foram selecionadas diversas tecnologias e ferramentas específicas para garantir uma infraestrutura robusta e eficiente. Como sistema operacional, optou-se pelo Windows, uma escolha comum que oferece suporte para uma variedade de aplicativos e serviços, além de uma interface familiar e amplamente utilizada.

Para publicar o site na Internet, utilizamos a plataforma GitHub adicionando o site a um repositório público, que contém todos os arquivos do site. Configuramos o Source (configuração do GitHub que determina de onde os arquivos para o site do GitHub Pages devem ser servidos), selecionando a opção *Deploy from a Branch* (Implantar de uma Branch). Na seção do Branch (a branch selecionada determina qual conteúdo será publicado na página do GitHub), escolhemos a principal (Main) e a pasta raiz (root). Após salvar essas configurações, o sistema inicia a construção do site e, ao ser concluída essa construção, um domínio é disponibilizado para acessar o site.

Para o servidor web, foi adotado o WAMP Server, uma solução que inclui o Apache, MySQL e PHP, fornecendo um ambiente completo para o desenvolvimento e



hospedagem de aplicativos web. O Apache é um servidor web amplamente utilizado, conhecido por sua confiabilidade e desempenho. O MySQL é um sistema de gerenciamento de banco de dados relacional, ideal para armazenar e gerenciar os dados do sistema de academia (INTELLIJ IDEA, 2024). O PHP é uma linguagem de script do lado do servidor que permite a criação de páginas web dinâmicas e interativas. Além disso, considerando a necessidade de acesso fácil e seguro aos sistemas e informações da academia, foi recomendada a utilização de uma rede em nuvem, com destaque para a Amazon Web Services (AWS).

A AWS oferece uma plataforma completa e flexível, com uma ampla gama de serviços, incluindo computação em nuvem, armazenamento, banco de dados, segurança e muito mais. Utilizar a AWS proporcionaria escalabilidade, alta disponibilidade, segurança avançada e fácil integração com outras ferramentas e serviços necessários para o projeto.

Dessa forma, a combinação do sistema operacional Windows, do servidor web WAMPServer e da utilização da AWS como plataforma em nuvem oferece uma infraestrutura sólida e completa para atender às necessidades de conectividade, segurança e flexibilidade do sistema de academia proposto.

## **5.1 Resumo do Capítulo**

Resumidamente, a combinação de uma topologia de rede hierárquica local com a utilização de serviços em nuvem, como a AWS, seria uma escolha ideal para atender às necessidades de conectividade, segurança e flexibilidade do sistema de academia proposto.

## 6 ANÁLISE DO IMPACTO NO NEGÓCIO

### 6.1 Sucesso do Projeto

O projeto de desenvolvimento do sistema de academia foi um sucesso em diversos aspectos:

**Atender às necessidades dos usuários:** O sistema oferece funcionalidades que atendem às necessidades dos alunos, professores e administradores da academia, facilitando o agendamento de aulas, registro de presença, gerenciamento de horários e acesso a informações relevantes.

**Usabilidade e experiência do usuário:** A interface do sistema é intuitiva e fácil de usar, sendo acessível em diferentes dispositivos, como computadores, tablets e smartphones.

**Segurança e confiabilidade:** O sistema implementa medidas de segurança adequadas para proteger as informações dos usuários e garantir a confiabilidade das transações financeiras.

**Desempenho e escalabilidade:** O sistema é capaz de lidar com um grande número de usuários simultâneos e pode ser facilmente escalado para atender a um crescimento na demanda.

**Manutenção e portabilidade:** O código do sistema é bem estruturado e documentado, facilitando a manutenção e atualizações futuras. O sistema também é portátil e pode ser executado em diferentes ambientes.

### 6.2 Feedback da Empresa

A empresa destaca os seguintes pontos positivos:

**Facilidade de uso:** O sistema é muito fácil de usar pelos alunos, professores e administradores.

**Funcionalidades completas:** O sistema oferece todas as funcionalidades que a empresa precisava.

**Segurança e confiabilidade:** A empresa está confiante de que as informações dos usuários estão seguras e que o sistema é confiável.

**Suporte técnico:** A equipe de desenvolvimento do sistema está sempre disponível para fornecer suporte técnico.

### 6.3 Dificuldades Enfrentadas

Algumas dificuldades foram enfrentadas durante o desenvolvimento do projeto, mas foram superadas com sucesso:

**Gerenciamento de tempo e recursos:** O projeto foi desenvolvido em um prazo apertado com recursos limitados.

**Garantir a qualidade do código:** A equipe de desenvolvimento precisou trabalhar duro para garantir que o código do sistema fosse de alta qualidade.

## 7 CONCLUSÃO

Este trabalho apresentou um sistema completo para gestão de uma academia com base nos requisitos funcionais e não funcionais detalhados neste documento, foi desenvolvido um sistema completo para gestão de uma academia, abrangendo desde o cadastro de usuários até a gestão de horários de aula e controle de presença. A utilização de tecnologias como MySQL, Spring Boot e HTML/CSS/JS proporcionou uma solução robusta e escalável, capaz de atender às demandas dos usuários e stakeholders.

Durante o processo de desenvolvimento, enfrentamos desafios significativos, como a gestão de tempo e recursos limitados, bem como a garantia da qualidade do código. No entanto, por meio do trabalho árduo da equipe e do apoio mútuo, conseguimos superar esses obstáculos e entregar um produto final de alta qualidade.

A validação do sistema por meio de um estudo de viabilidade demonstrou que todas as funcionalidades propostas foram implementadas com sucesso e atendem aos requisitos estabelecidos. Além disso, o feedback da empresa e dos usuários tem sido positivo, destacando a facilidade de uso, a segurança e a confiabilidade do sistema.

Em suma, o projeto de desenvolvimento do sistema de academia foi bem-sucedido em fornecer uma solução abrangente e eficaz para as necessidades dos usuários e da empresa. Este trabalho servirá como base para futuras melhorias e expansões, garantindo que a academia possa continuar oferecendo uma experiência satisfatória aos seus alunos e colaboradores.

## REFERÊNCIAS

DBEAVER. **Ferramenta de gerenciamento de banco de dados multiplataforma.** Disponível em: <https://dbeaver.com/> . Acesso em: 03 abr. 2024.

GITHUB. **Plataforma para hospedagem e compartilhamento de código-fonte.** Disponível em: <https://github.com/> . Acesso em: 04 abr. 2024.

INTELLIJ IDEA. **IDE para desenvolvimento Java.** Disponível em: <https://www.jetbrains.com/idea/> . Acesso em: 05 abr. 2024.

JAVA SPRING. **Documentação oficial do Spring Framework.** Disponível em: <https://start.spring.io/> . Acesso em: 05 abr. 2024.

MDN WEB DOCS. **Documentação oficial das tecnologias web.** Disponível em: <https://developer.mozilla.org/> . Acesso em: 06 abr. 2024.

MAVEN. **Ferramenta de gerenciamento de dependências para projetos Java.** Disponível em: <https://maven.apache.org/install.html> . Acesso em: 06 abr. 2024.

MYSQL DOCUMENTATION. **Documentação oficial do MySQL.** Disponível em: <https://dev.mysql.com/doc/> . Acesso em: 06 abr. 2024.

SQL TUTORIAL. **Tutoriais e referências para SQL.** Disponível em: <https://www.w3schools.com/sql/> . Acesso em: 09 abr. 2024.

STACK OVERFLOW. **Fórum de perguntas e respostas para desenvolvedores.** Disponível em: <https://pt.stackoverflow.com/> . Acesso em: 09 abr. 2024.

W3SCHOOLS. **Tutoriais e referências para HTML, CSS, JavaScript e outras tecnologias web.** Disponível em: <https://www.w3schools.com/> . Acesso em: 09 abr, 2024.

### APÊNDICE A - EQUIPE

Nome	Função	Tipo de Desenvolvimento	Matrícula
Alan Hugo Lins Barbosa	Desenvolvedor	Back	2023281175
Cláudio José do Espírito Santo Junior	Testador	-	2023276228
Cleyandson Gustavo Caetano de Lima Fragoso da Silva	Lógico	-	2023175804
Gabriel Lucas Silva	Analista de Requisitos	-	2023278970
Gabriel Machado Brilhante	Testador	-	2023278856
Gabriel Victor Araújo Vieira	Desenvolvedor	Front	2021135249
Heleno José da Silva Junior	Desenvolvedor	Front	2023171605
Izabelly Pinto Rodrigues	Desenvolvedora	Back	2023283351
João Vitor da Silva	Testador	-	2023163087
Kaique Genuino da Paz Vilela	Analista de Requisitos	-	2018108137
Lukas Rodrigues da Silva	Desenvolvedor	Front	2023278620
Leandro Ribeiro	Conceitual	-	2023276919

Mateus Paz de Oliveira	Físico	-	2023176015
Matheus Ferro Araujo	Lógico	-	2023162905
Mirella Patrícia dos Santos Bispo	Conceitual	-	2023282709
Natan José de Araújo	Analista de Requisitos / Físico	-	2023277966
Thaís Mirelle Silva	Analista de Requisitos	-	2023283091
Victor Alexandre Hermínio de Lima	Analista de Requisitos	-	2023278378
Vitor Manoel Gomes de Santana França	Lógico	-	2022254998
Willian Silva Nunes	Analista de Requisitos	-	2023163741