**SenacTech**

**Autor(es):** Artur Villanova,

Igor Rangel,

João Gusmão,

Ricardo Abreu,

Marcos Biederman,

Mauricio T Welter

**Desenvolvimento de um Ambiente Virtual de Aprendizado**

**Cidade: Porto Alegre**

**Ano: 2024**

Artur Villanova, Igor Rangel, João Gusmão, Ricardo Abreu, Marcos Biederman, Mauricio T Welter

**Desenvolvimento de um Ambiente Virtual de Aprendizado**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico de Desenvolvimento de Sistemas, da Instituição SenacTech, como requisito parcial para obtenção do Certificado de Técnico em Desenvolvimento de Sistemas. - Senac Tech

**Orientador:** Professor Gilmar Aires dos Santos

**Porto Alegre**

**2024**

**Sumário**

[1 Introdução 1](#_heading=h.gjdgxs)

[2 Objetivos 2](#_heading=h.30j0zll)

[2.1 Objetivo Geral 2](#_heading=h.1fob9te)

[2.2 Objetivos Específicos 2](#_heading=h.3znysh7)

[3 Levantamento, Especificação e Análise de Requisitos 4](#_heading=h.2et92p0)

[3.1 Requisitos Funcionais 4](#_heading=h.tyjcwt)

[3.2 Requisitos Não Funcionais 5](#_heading=h.3dy6vkm)

[4 Diagrama UML 8](#_heading=h.1t3h5sf)

[4.1 Caso de Uso 9](#_heading=h.4d34og8)

[4.1.1 Administrador 9](#_heading=h.2s8eyo1)

[4.1.2 Professor 11](#_heading=h.17dp8vu)

[4.1.3 Aluno. 14](#_heading=h.3rdcrjn)

[4.2 Diagrama de Classe 16](#_heading=h.26in1rg)

[4.3 Diagrama de Sequência 17](#_heading=h.lnxbz9)

[4.3.1 Administrador 17](#_heading=h.35nkun2)

[4.3.2 Professor 19](#_heading=h.44sinio)

[4.3.3 Aluno 22](#_heading=h.2jxsxqh)

[5 Banco de Dados 23](#_heading=h.z337ya)

[5.1 Entidade 23](#_heading=h.3j2qqm3)

[5.2 Dicionário de dados 24](#_heading=h.1y810tw)

[5.3 DDL 25](#_heading=h.4i7ojhp)

[5.3.1 Código do Banco 26](#_heading=h.2xcytpi)

[6 Telas do Projeto e suas Definições 30](#_heading=h.1ci93xb)

[7 Engenharia de Testes 35](#_heading=h.3whwml4)

[7.1 Teste de Conexão com o Banco de Dados 35](#_heading=h.2bn6wsx)

[7.1.1 Configuração do Ambiente 35](#_heading=h.qsh70q)

[7.1.2 Script de Teste 35](#_heading=h.3as4poj)

[7.1.3 Execução do Teste 36](#_heading=h.1pxezwc)

[7.2 Teste de Integridade do Banco de Dados 36](#_heading=h.49x2ik5)

[7.2.1 Verificação das Chaves Estrangeiras 36](#_heading=h.2p2csry)

[7.2.2 Validação dos Dados 36](#_heading=h.147n2zr)

[7.2.3 Teste de CRUD 37](#_heading=h.3o7alnk)

[7.3 Testes de Frontend 37](#_heading=h.23ckvvd)

[7.3.1 Teste de Interface do Usuário 37](#_heading=h.ihv636)

[7.3.2 Teste de Responsividade 37](#_heading=h.1hmsyys)

[7.4 Testes de Backend 38](#_heading=h.41mghml)

[7.4.1 Teste de Rotas e Requisições HTTP 38](#_heading=h.2grqrue)

[7.4.2 Teste de Lógica de Negócios 38](#_heading=h.vx1227)

[7.4.3 Teste de Integração 39](#_heading=h.3fwokq0)

[7.4.4 Teste de Desempenho 39](#_heading=h.1v1yuxt)

[8 Conclusão 41](#_heading=h.2u6wntf)

[9 Referências 42](#_heading=h.19c6y18)

[10 Anexos 43](#_heading=h.3tbugp1)

[10. 1 Telas dos códigos 43](#_heading=h.28h4qwu)

[10.1.1 Administrador 43](#_heading=h.nmf14n)

[10.1.2 Descrição de algumas funções no código do Administrador 45](#_heading=h.37m2jsg)

[10.2 Professor 46](#_heading=h.1mrcu09)

[10.2.1 Descrição de algumas funções no código do Professor 47](#_heading=h.46r0co2)

[10.3 Aluno 48](#_heading=h.2lwamvv)

[10.3.1 Descrição de algumas funções no código do Aluno 51](#_heading=h.111kx3o)

[10.4 Diagrama de Classe 51](#_heading=h.3l18frh)

**Resumo**

O projeto Eduq é uma iniciativa voltada para o desenvolvimento de um sistema educacional abrangente e eficaz, destinado a facilitar a gestão acadêmica e promover uma experiência de aprendizado intuitiva e eficiente para alunos, professores e administradores. Com uma abordagem cuidadosa e meticulosa, cada aspecto do sistema foi concebido e implementado com o objetivo de atender às necessidades variadas dos usuários e proporcionar uma plataforma educacional robusta e adaptável.

A documentação apresentada oferece uma visão detalhada e abrangente do projeto Eduq, desde a concepção inicial até o desenvolvimento de um MVP do sistema. Compreende diversos aspectos, incluindo o modelo conceitual do banco de dados, a arquitetura do sistema, os testes de frontend e backend, além de abordar temas como engenharia de testes e integração de componentes.

Na introdução, destaca-se a importância da documentação como um guia fundamental para o desenvolvimento e implementação do sistema, fornecendo uma estrutura clara e abrangente para todas as etapas do processo. Através dessa documentação, busca-se não apenas oferecer um registro detalhado do sistema, mas também destacar a importância da engenharia de testes e garantir a integridade, desempenho e eficácia do sistema proposto.

A conclusão reforça a relevância da documentação como um investimento estratégico para o cliente, destacando seu valor na mitigação de riscos, aceleração do processo de desenvolvimento e garantia da entrega de um produto final de alta qualidade e desempenho. Além disso, ressalta-se o papel da documentação como um recurso valioso para treinamento e capacitação de novos membros da equipe, garantindo uma transição suave e eficiente durante o ciclo de vida do projeto.

Em suma, a documentação apresentada representa não apenas um registro do sistema educacional proposto, mas também um compromisso com a excelência, transparência e eficácia do projeto como um todo. Ao investir na criação dessa documentação, o cliente estabelece as bases para o sucesso e a evolução contínua do sistema Eduq, promovendo uma experiência educacional enriquecedora e satisfatória para todos os envolvidos.

# 1. Introdução

A implementação de um sistema educacional eficaz requer uma abordagem cuidadosa e meticulosa em todas as etapas do processo de desenvolvimento. Desde a concepção inicial até a entrega final, é essencial garantir que cada componente do sistema funcione harmoniosamente junto, atendendo às necessidades dos usuários de forma confiável e eficiente. Nesse contexto, a documentação desempenha um papel fundamental, fornecendo uma estrutura clara e abrangente para o desenvolvimento, teste e implementação do sistema.

A documentação apresentada aqui abrange todos os aspectos do projeto de um sistema educacional, desde o modelo conceitual do banco de dados até os testes de frontend e backend. Cada seção foi cuidadosamente elaborada para fornecer uma visão detalhada das funcionalidades do sistema, das interações dos usuários e dos procedimentos de teste e validação.

Ao longo deste documento, você encontrará diagramas de sequência que ilustram as interações entre professores, alunos e o sistema, modelos conceituais e lógicos do banco de dados que definem a estrutura de armazenamento de informações, e descrições detalhadas dos testes realizados para garantir a integridade e o desempenho do sistema.

Por meio dessa documentação abrangente, buscamos fornecer uma base sólida para o desenvolvimento e implementação bem-sucedidos de um sistema educacional que atenda às necessidades dos usuários e promova uma experiência de aprendizado eficaz e intuitiva.

Além disso, a documentação destaca a importância da engenharia de testes em todas as fases do projeto. Os testes abrangem desde a verificação da integridade do banco de dados até a validação da interface do usuário e da lógica de negócios no backend. Essa abordagem rigorosa de testes garante que o sistema seja robusto, confiável e capaz de lidar com uma variedade de cenários de uso, proporcionando uma experiência consistente e satisfatória para os usuários finais.

Por fim, a documentação não apenas fornece uma visão abrangente do sistema educacional em questão, mas também serve como um guia valioso para o desenvolvimento futuro e aprimoramento contínuo do sistema.

# 2. Objetivos

Este sistema o qual estamos desenvolvendo surgiu após conversarmos e ouvirmos diversas opiniões de diferentes alunos do Senac Tech e de outras instituições de ensino, e uma grande maioria reclamava do AVA utilizado por estas instituições, de modo que havia pontos em comuns entre eles, mesmo sendo de diferentes instituições. Falamos também com alguns professores e os mesmos tinham suas ressalvas sobre a plataforma e do modo de lidar com ela. Então após ouvirmos as reclamações dos alunos e dos professores, e nós mesmos sendo alunos e sabendo os problemas que muitos têm com o uso da plataforma, decidimos então criar um AVA mais intuitivo e fácil de utilizar, tanto pelos alunos quanto pelos professores.

O primeiro desafio ao criar nosso sistema, ou AVA, foi encontrar um nome para o mesmo, novamente como equipe nos juntamos e começamos a fazer o brainstorm, onde começamos a jogar ideias e juntar opiniões sobre como gostaríamos que o sistema fosse, e quais seriam os requisitos que seriam necessários para o sistema ser acessível e bem aceito pelos usuários que talvez virão a utilizar e testar nossa plataforma em um futuro próximo.

Após o brainstorm decidimos que o nome da nossa plataforma seria Eduq, um nome que demonstra o objetivo do nosso AVA em ajudar e facilitar o acesso dos usuários para que a jornada de aprendizagem dos mesmos seja agradável e produtiva, e que de fato faça a diferença para os mesmos. Com o nome definido começamos então a pensar mais a fundo em quais eram os nossos objetivos ao desenvolver este sistema, e os definimos da seguinte forma:

## 2.1 Objetivo Geral

Desenvolver um Ambiente Virtual de Aprendizado (AVA) chamado Eduq.

## 2.2 Objetivos Específicos

* **Criar um sistema de cadastro e autenticação seguro:** Implementar um mecanismo robusto de cadastro para professores e garantir uma autenticação segura baseada em credenciais, visando proteger os dados pessoais dos usuários.
* **Desenvolver um sistema:** Que seja intuitivo, acessível e eficiente, visando melhorar a experiência de ensino e aprendizagem para alunos e professores de instituições educacionais.
* **Facilitar o gerenciamento de conteúdo:** Desenvolver uma interface intuitiva que permita aos professores realizar upload, atualização e organização de materiais de curso de forma eficiente, incluindo a capacidade de transcrição de vídeos para texto e controle de versões dos materiais.
* **Possibilitar a criação e entrega de atividades interativas:** Criar ferramentas que permitam aos professores elaborar atividades interativas com perguntas relevantes, com opção de conversão de pontos em notas e acompanhamento do desempenho dos alunos, além de um sistema para os alunos enviarem suas respostas e trabalhos.
* **Promover avaliação e feedback eficazes:** Desenvolver uma ferramenta de avaliação de atividades dos alunos pelos professores, com visualização de notas e feedback pelos alunos, garantindo registro histórico das avaliações.
* **Facilitar o agendamento e organização de eventos:** Implementar um calendário que exibe datas relevantes, como entregas de atividades e aulas ao vivo, além de possibilitar o agendamento de sessões de tutoria pelos professores e alunos.
* **Fomentar a interação por meio de fóruns de discussão:** Criar fóruns de discussão para cada curso, moderados por IA, para facilitar a comunicação entre alunos e professores.
* **Oferecer suporte por meio de chat com IA:** Integrar um chat com IA para responder dúvidas simples dos alunos em tempo real, gerando relatórios de uso para análise.
* **Garantir níveis de acesso e segurança adequados:** Definir quatro níveis de acesso com permissões restritas, registrar atividades e auditar acessos para garantir a segurança dos dados.
* **Facilitar sessões de dúvidas individuais:** Permitir que alunos agendem sessões de dúvidas individuais com professores, com acompanhamento do status das sessões.
* **Assegurar a publicação segura de provas e avaliações:** Possibilitar o upload seguro de exames e avaliações pela equipe administrativa, com controle de acesso aos materiais.

Com isso os objetivos do nosso projeto foram estabelecidos, e começamos então a concentrar nossos esforços em reunir materiais e referências para a próxima parte do projeto. Essa fase inicial foi crucial para que pudéssemos estabelecer uma base sólida no desenvolvimento do nosso projeto. Enquanto trabalhávamos na estrutura inicial, também avançávamos na definição dos requisitos funcionais e não funcionais, assim como das regras de negócio, fundamentais para orientar nosso trabalho. Esses aspectos detalhados serão abordados de forma mais ampla no próximo capítulo.

# 3. Levantamento, Especificação e Análise de Requisitos

## 3.1 Requisitos Funcionais

**Cadastro e Autenticação:**

- Sistema de cadastro de professores pelo administrativo.

- Mecanismo seguro de autenticação baseado em credenciais (login e senha).

- Recuperação de senha para usuários.

**Gerenciamento de Conteúdo:**

- Sistema intuitivo para professores fazerem upload, atualizarem e organizarem materiais de curso.

- Capacidade de transcrição eficiente de vídeos para texto.

- Controle de versões e histórico de alterações nos materiais.

**Criação e Entrega de Atividades:**

- Capacidade para professores criarem atividades interativas com perguntas relevantes.

- Opção de conversão de pontos em notas, seguindo uma escala predefinida.

- Interface para acompanhamento de desempenho individual e geral.

- Sistema de envio de respostas e trabalhos por parte dos alunos.

**Avaliação e Feedback:**

- Ferramenta de avaliação de atividades dos alunos por parte dos professores.

- Visualização de notas e feedback pelos alunos.

- Registro do histórico de notas e feedback

**Agendamento e Calendário:**

- Calendário exibindo datas relevantes, como entrega de atividades, aulas ao vivo e avaliações.

- Possibilidade de professores e alunos agendarem sessões de tutoria.

**Fórum de Discussão:**

- Fórum de discussão para cada curso.

- IA moderadora para manter a comunicação adequada.

**Suporte de IA e Chat:**

- Integração de um chat com IA para responder dúvidas simples dos alunos em tempo real.

- Geração de relatórios de uso do chat.

**Níveis de Acesso e Segurança:**

- Definição de quatro níveis de acesso (aluno, professor, administrativo, pedagógico) com permissões restritas.

- Registo de atividades e auditoria de acesso.

**Sessões de Dúvidas e Tutoria:**

- Possibilidade de alunos agendarem sessões de dúvidas individuais com professores.

- Acompanhamento do status das sessões de dúvidas.

**Publicação de Provas e Avaliações:**

- Possibilidade de a equipe administrativa fazer o upload de exames e avaliações.

- Garantia de acesso controlado aos materiais de avaliação.

## 3.2 Requisitos Não Funcionais

## **Segurança:**

- Garantir a segurança dos dados pessoais dos usuários.

- Implementar protocolos de segurança para proteger contra ameaças cibernéticas.

- Realizar auditorias de segurança regulares.

**Acessibilidade:**

- Garantir que o AVA seja acessível a todos, incluindo pessoas com deficiência.

- Conformidade com padrões de acessibilidade, como WCAG.

**Desempenho:**

- Garantir um desempenho rápido e responsivo do sistema, mesmo em horários de pico.

- O tempo de resposta das páginas não deve exceder um limite especificado.

**Escalabilidade:**

- O sistema deve ser escalável para lidar com um aumento no número de usuários.

- Deve ser possível adicionar servidores ou recursos conforme necessários.

**Conformidade Legal:**

- Cumprir regulamentos legais relacionados à educação e proteção de dados.

- Manter registros e relatórios para fins de conformidade.

**Usabilidade:**

- Interface de usuário intuitiva e amigável para facilitar o uso por alunos, professores e administradores.

- Testes de usabilidade regulares para otimização da experiência do usuário.

**Disponibilidade:**

- Garantir alta disponibilidade do sistema, com tempo de inatividade mínimo planejado.

- Implementar procedimentos de backup e recuperação de desastres.

**Integração com Sistemas Externos:**

- Integração com sistemas de gestão acadêmica, sistemas de autenticação, e outras ferramentas de suporte.

- Garantir que os dados sejam consistentes entre sistemas.

**Internacionalização:**

- Suporte a múltiplos idiomas e culturas, se aplicável.

- Garantir que datas, formatos de números e moedas sejam adaptáveis.

**Manutenção e Suporte:**

- Plano de manutenção regular, atualizações de software e suporte técnico contínuo.

- Disponibilidade de suporte técnico para usuários.

Após a etapa de levantamento de requisitos e do desenvolver das regras de negócio, a base de nosso projeto estava pronta, e a partir daí começamos então a fazer os UML, nesta parte do projeto nos reunimos para definir como seria nosso sistema, quais seriam os passos que os usuários teriam que dar para chegar nas nossas telas, desde o acesso inicial, do login e do uso das features que virão a estar presente no AVA em que logo entraria em sua fase de desenvolvimento.

O nosso sistema será estruturado com base em diferentes níveis de acesso, cada um com suas respectivas permissões e funcionalidades. Primeiramente, temos o nível de administrador, que desfrutará de privilégios abrangentes dentro do sistema. Este usuário terá total controle sobre as operações, podendo criar e gerenciar cursos, turmas e atribuir professores às respectivas turmas. Além disso, o administrador terá a capacidade de efetuar qualquer tipo de alteração no sistema, garantindo assim sua flexibilidade e adaptabilidade às necessidades em constante evolução.

O nível de acesso Professor será destinado aos educadores, oferecendo acesso às turmas que lhes forem atribuídas. Dentro dessas turmas, terão a capacidade de disponibilizar materiais didáticos para os alunos, responder às perguntas dos estudantes, participar dos fóruns de discussão e atribuir notas às atividades realizadas. Essas funcionalidades permitirão aos professores uma gestão eficiente do ensino e interação direta com os alunos.

Por fim, temos o nível de acesso Aluno, que será destinado aos estudantes matriculados nos cursos. Os alunos terão acesso às aulas ministradas em suas turmas, aos materiais disponibilizados pelos professores, aos fóruns de discussão para interação com colegas e professores, além da possibilidade de tirar dúvidas diretamente com os professores e verificar as notas atribuídas às suas atividades. Essa estrutura proporcionará aos alunos uma experiência de aprendizado interativa e personalizada, promovendo assim o seu engajamento e sucesso acadêmico.

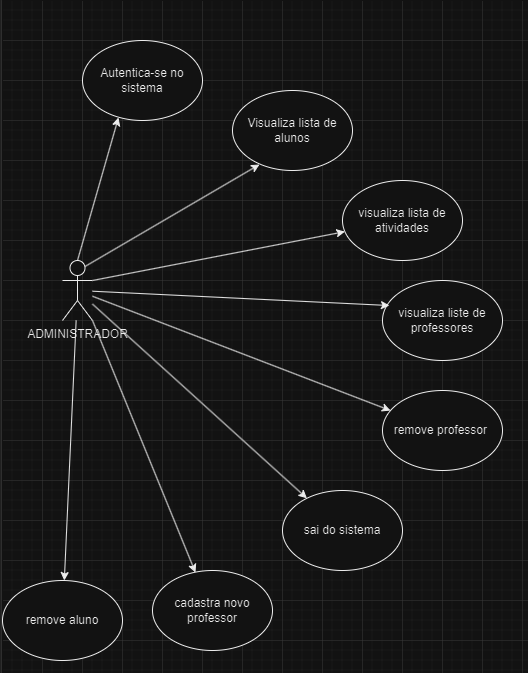
Com os níveis de acesso definidos, iniciamos o desenvolvimento do diagrama UML para o nosso sistema, o Eduq. Para garantir a precisão e eficácia da diagramação, reunimos referências pertinentes e buscamos orientação junto ao nosso orientador. Esta etapa foi fundamental para estabelecer a estrutura e as relações entre os diferentes elementos do sistema, proporcionando uma visão abrangente de sua arquitetura e funcionalidades. O processo de criação do diagrama UML foi conduzido com cuidado e atenção aos detalhes, visando representar de forma clara e concisa todas as características e interações do Eduq.

# 4. Diagrama UML

Nesta parte do trabalho será abordada a diagramação de nosso projeto, incluindo a explicação e a aplicação prática de diversos tipos de diagramas essenciais para a modelagem do sistema. Serão detalhados o diagrama de caso de uso, que descreve as interações entre os usuários e o sistema; o diagrama de classes, que ilustra a estrutura estática do sistema através das classes, seus atributos e relacionamentos; e o diagrama de sequência, que representa a interação entre os objetos ao longo do tempo, detalhando a ordem das mensagens trocadas.

## 4.1 Caso de Uso

### 4.1.1 Administrador

****Figura 1 - Caso de uso do Administrador.

Fonte: Autor

#### Descrição Geral

O administrador do sistema é responsável por gerenciar usuários e atividades dentro da plataforma. Ele possui permissões para autenticar-se, visualizar listas de alunos, professores e atividades, além de realizar a gestão de cadastros e remoções de professores e alunos. Por fim, o administrador deve encerrar sua sessão com segurança através do log out.

#### Fluxo Principal

**Autenticação no Sistema:**

* + O administrador acessa a página de login do sistema.
  + Ele insere suas credenciais (nome de usuário e senha) e confirma.
  + O sistema valida as credenciais e, se estiverem corretas, concede acesso ao administrador, redirecionando-o para o painel principal.

**Visualização da Lista de Alunos:**

* + No painel principal, o administrador seleciona a opção "Lista de Alunos".
  + O sistema exibe uma tabela com todos os alunos cadastrados, incluindo detalhes como nome, matrícula, curso e status.

**Visualização da Lista de Atividades:**

* + O administrador seleciona a opção "Lista de Atividades".
  + O sistema apresenta uma lista detalhada das atividades cadastradas, incluindo título, descrição, data de criação, e responsáveis.

**Visualização da Lista de Professores:**

* + O administrador seleciona a opção "Lista de Professores".
  + O sistema exibe uma lista completa dos professores, com informações como nome, disciplina(s) ministrada(s), e contato.

**Remoção de Professores:**

* + O administrador escolhe um professor na lista e seleciona a opção "Remover".
  + O sistema solicita uma confirmação da remoção.
  + Após a confirmação, o sistema exclui o professor do banco de dados e atualiza a lista.

**Cadastro de Novos Professores:**

* + O administrador acessa a opção "Cadastrar Novo Professor".
  + Ele preenche o formulário de cadastro com as informações necessárias (nome, disciplina, e-mail, etc.).
  + O sistema valida e salva as informações no banco de dados, adicionando o novo professor à lista.

**Cadastro de Novos Alunos:**

* + O administrador seleciona a opção "Cadastrar Novo Aluno".
  + Ele preenche o formulário de cadastro com as informações do aluno (nome, matrícula, curso, e-mail, etc.).
  + O sistema valida e salva os dados, adicionando o novo aluno à lista.

**Log Out:**

* + O administrador escolhe a opção "Log Out" no painel principal.
  + O sistema encerra a sessão do administrador e redireciona-o para a página de login, garantindo que o acesso seja devidamente fechado

#### Exceções

**Autenticação Falha:**

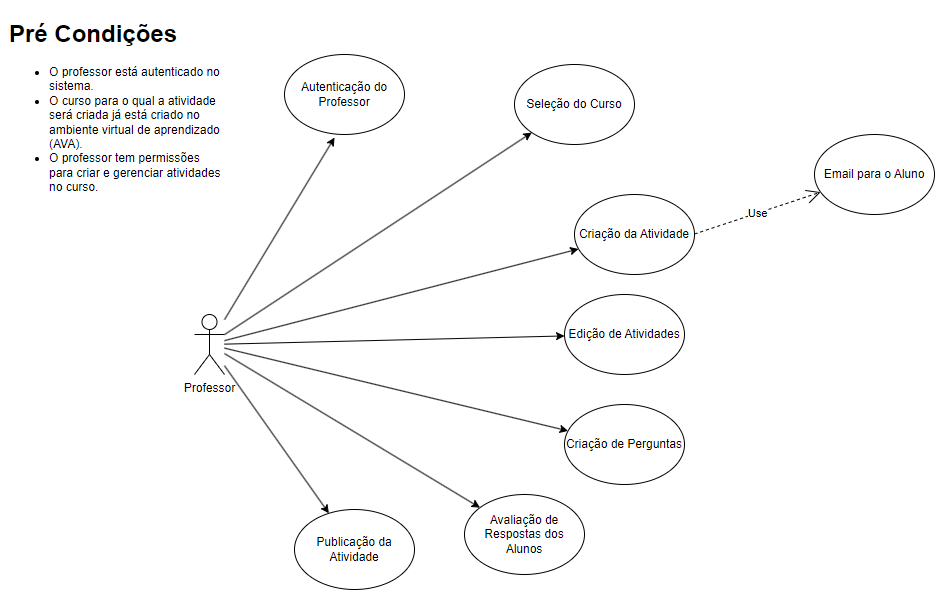
* + Se o administrador inserir credenciais inválidas, o sistema exibe uma mensagem de erro e solicita nova tentativa.

**Erro na Remoção/Cadastro:**

* + Caso ocorra um erro ao tentar remover ou cadastrar um usuário, o sistema exibe uma mensagem de erro detalhada e sugere ações corretivas.

Esta história do caso de uso ilustra as principais interações e operações realizadas pelo administrador no sistema, garantindo a gestão eficiente de alunos, professores e atividades.

### 4.1.2 Professor

Imagem 2 - Caso de uso do Professor.

Fonte: Autor

#### Descrição Geral

O professor é responsável por criar e gerenciar atividades para os alunos, além de responder dúvidas relacionadas às disciplinas que leciona. Ele possui permissões para autenticar-se no sistema, selecionar cursos, criar e editar atividades, responder perguntas dos alunos, e verificar a avaliação de suas respostas e a submissão das atividades pelos alunos. Por fim, o professor deve encerrar sua sessão com segurança através do log out.

#### Fluxo Principal

**Autenticação no Sistema:**

* + O professor acessa a página de login do sistema.
  + Ele insere suas credenciais (nome de usuário e senha) e confirma.
  + O sistema valida as credenciais e, se estiverem corretas, concede acesso ao professor, redirecionando-o para o painel principal.

**Seleção de Curso:**

* + No painel principal, o professor seleciona a opção "Cursos".
  + O sistema exibe a lista de cursos nos quais o professor foi registrado pelo administrador.
  + O professor escolhe um dos cursos para gerenciar.

**Criação de Atividade:**

* + O professor acessa a opção "Criar Atividade" dentro do curso selecionado.
  + Ele preenche o formulário de criação de atividade com título, descrição, data de entrega, e outros detalhes necessários.
  + O sistema salva a atividade e envia um e-mail automático aos alunos do curso com os detalhes da nova atividade.

**Edição de Atividade:**

* + Caso o professor perceba algum erro na atividade criada, ele seleciona a opção "Editar Atividade".
  + O sistema exibe a lista de atividades e o professor escolhe a atividade que deseja editar.
  + Ele faz as correções necessárias e salva as alterações.
  + O sistema atualiza a atividade e notifica os alunos sobre as mudanças via e-mail.

**Respostas a Perguntas dos Alunos:**

* + O professor acessa a seção "Dúvidas dos Alunos".
  + O sistema apresenta as perguntas postadas pelos alunos referentes à matéria do curso selecionado.
  + O professor lê as perguntas e responde conforme necessário.

**Verificação de Avaliação das Respostas:**

* + Após responder as perguntas, o professor verifica a sessão "Avaliações das Respostas".
  + O sistema mostra as avaliações dos alunos sobre as respostas do professor, indicando se as respostas foram úteis ou não.
  + O professor pode usar esse feedback para melhorar futuras respostas.

**Verificação de Submissão de Atividades:**

* + O professor acessa a seção "Submissões de Atividades".
  + O sistema exibe uma lista das atividades submetidas pelos alunos.
  + O professor verifica quais alunos já postaram as atividades e pode acessar os detalhes de cada submissão.

**Log Out:**

* + O professor escolhe a opção "Log Out" no painel principal.
  + O sistema encerra a sessão do professor e redireciona-o para a página de login, garantindo que o acesso seja devidamente fechado.

#### Exceções

**Autenticação Falha:**

* + Se o professor inserir credenciais inválidas, o sistema exibe uma mensagem de erro e solicita nova tentativa.

**Erro na Criação/Edição de Atividade:**

* + Caso ocorra um erro ao tentar criar ou editar uma atividade, o sistema exibe uma mensagem de erro detalhada e sugere ações corretivas.

**Erro ao Responder Perguntas:**

* + Se houver um problema ao responder perguntas ou ao verificar avaliações, o sistema notifica o professor e sugere uma nova tentativa ou contato com o suporte técnico.

Esta história do caso de uso ilustra as principais interações e operações realizadas pelo professor no sistema, garantindo a criação e gestão eficiente de atividades, além de um suporte eficaz aos alunos através das respostas às suas dúvidas.

### 4.1.3 Aluno

### Imagem 3 - Caso de uso do Aluno.

Fonte: Autor

#### Descrição Geral

O aluno é responsável por acessar materiais do curso, realizar atividades, verificar notas, solicitar tutorias, postar perguntas aos professores, e participar de discussões no fórum com colegas. Ele possui permissões para autenticar-se no sistema, visualizar materiais do curso, acessar atividades, verificar suas notas, interagir com os professores e colegas, e por fim, encerrar sua sessão com segurança através do log out.

#### Fluxo Principal

**Acesso ao Sistema:**

* + O aluno acessa a página de login do sistema.
  + Ele insere suas credenciais (nome de usuário e senha) e confirma.
  + O sistema valida as credenciais e, se estiverem corretas, concede acesso ao aluno, direcionando-o para o painel principal.

**Visualização dos Materiais do Curso:**

* + No painel principal, o aluno seleciona a opção "Materiais do Curso".
  + O sistema exibe uma lista de materiais disponíveis, como apostilas, slides, vídeos, e leituras recomendadas.
  + O aluno navega pelos materiais e acessa os que forem necessários para seus estudos.

**Seleção de Atividade:**

* + O aluno acessa a seção "Atividades" no painel do curso.
  + O sistema apresenta uma lista de atividades disponíveis para o curso.
  + O aluno seleciona uma das atividades para visualizar os detalhes, como a descrição, prazo de entrega e critérios de avaliação.

**Verificação de Notas:**

* + O aluno seleciona a opção "Notas" no painel principal.
  + O sistema exibe as notas obtidas pelo aluno em todas as atividades realizadas até o momento.
  + O aluno revisa suas notas e verifica seu desempenho no curso.

**Solicitação de Tutoria:**

* + O aluno acessa a seção "Tutoria" no painel principal.
  + Ele preenche um formulário de solicitação de tutoria, indicando o tema ou dúvida específica para a qual precisa de ajuda.
  + O sistema envia a solicitação ao professor responsável, que entrará em contato para agendar a tutoria.

**Postagem de Pergunta ao Professor:**

* + O aluno vai até a seção "Perguntas ao Professor".
  + Ele insere sua pergunta referente a algum tópico do curso ou atividade específica.
  + O sistema registra a pergunta e a encaminha ao professor, que poderá respondê-la na seção de dúvidas.

**Participação no Fórum:**

* + O aluno acessa a seção "Fórum" no painel principal.
  + Ele navega pelas discussões em andamento iniciadas por colegas.
  + O aluno seleciona uma discussão e participa, postando suas dúvidas, respostas ou comentários pertinentes ao tema em debate.

**Log Out:**

* + O aluno escolhe a opção "Log Out" no painel principal.
  + O sistema encerra a sessão do aluno e redireciona-o para a página de login, garantindo que o acesso seja devidamente fechado.

#### Exceções

**Autenticação Falha:**

* + Se o aluno inserir credenciais inválidas, o sistema exibe uma mensagem de erro e solicita nova tentativa.

**Erro ao Visualizar Materiais/Atividades:**

* + Caso ocorra um erro ao tentar acessar materiais do curso ou atividades, o sistema exibe uma mensagem de erro e sugere ações corretivas.

**Erro na Solicitação de Tutoria/Postagem de Pergunta:**

* + Se houver um problema ao solicitar tutoria ou postar perguntas, o sistema notifica o aluno e sugere uma nova tentativa ou contato com o suporte técnico.

**Erro no Fórum:**

* + Caso ocorra um problema ao participar do fórum, o sistema exibe uma mensagem de erro e sugere uma nova tentativa ou contato com o suporte técnico.

Esta história do caso de uso ilustra as principais interações e operações realizadas pelo aluno no sistema, garantindo um acesso eficiente aos materiais e atividades do curso, suporte adequado através de tutorias e perguntas, além de um engajamento ativo nas discussões do fórum.

## 4.2 Diagrama de Classe

O diagrama de classe para o nosso ambiente virtual de aprendizagem ilustra a estrutura estática do sistema, mostrando as classes principais (Administrador, Professor, Aluno), seus atributos, métodos, e os relacionamentos entre elas. Este diagrama é essencial para a modelagem do sistema, fornecendo uma visão clara das responsabilidades de cada classe e como elas interagem.

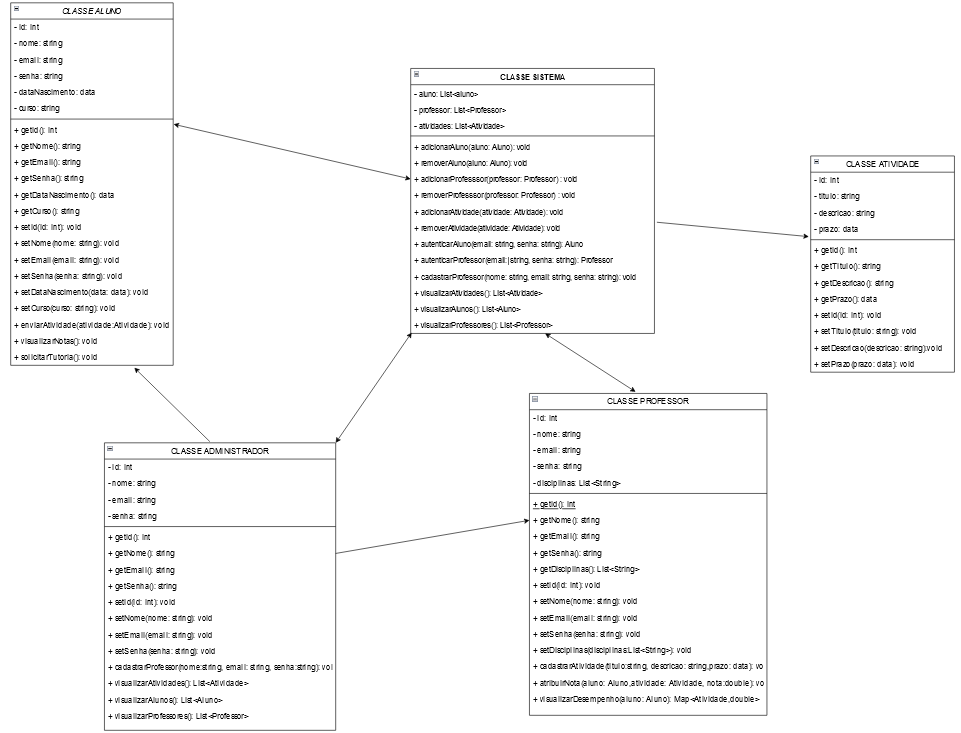


Imagem 4 - Diagrama de Classe.

Fonte: Autor

## 4.3 Diagrama de Sequência

### 4.3.1 Administrador

O diagrama de sequência do administrador detalha a interação entre o administrador e o sistema para realizar tarefas específicas, como autenticação, visualização de listas (alunos, atividades, professores), gerenciamento de professores e alunos (remoção e cadastro), gerenciar turmas (remover, listar, adicionar e editar), cadastro de novos alunos e finalização da sessão (log out). Este diagrama ajuda a visualizar o fluxo de eventos e a comunicação entre o administrador e o sistema.

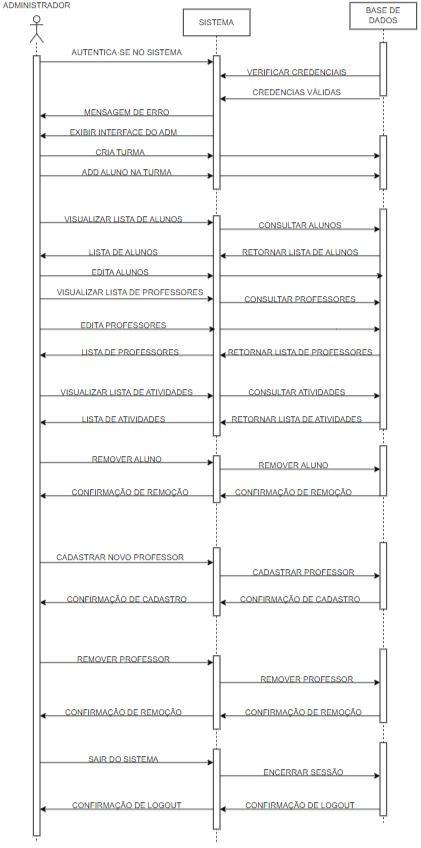


Imagem 5 - Diagrama de Sequência do Administrador.

Fonte: Autor

### Considerações

**Autenticação:** A autenticação é a primeira interação, garantindo que apenas usuários autorizados possam acessar as funcionalidades do sistema.

**Visualização de Listas:** O administrador pode visualizar listas de alunos, atividades e professores, permitindo uma visão geral e gerenciamento eficiente.

**Gerenciamento de Professores:** O administrador pode remover e cadastrar professores, mantendo a equipe docente atualizada.

**Cadastro de Alunos:** O administrador pode cadastrar novos alunos, assegurando que todos os alunos sejam devidamente registrados no sistema.

**Log Out:** O administrador encerra a sessão de forma segura, garantindo a proteção das informações do sistema.

Este diagrama de sequência fornece uma visão detalhada das interações do administrador com o sistema, facilitando a compreensão do fluxo de eventos e a comunicação entre os componentes do sistema.

### 4.3.2 Professor

O diagrama de sequência do professor ilustra as interações entre o professor e o sistema para realizar diversas atividades, desde a autenticação até a criação de atividades e interação com os alunos. Este diagrama detalha o fluxo de eventos e a comunicação necessária para que o professor possa executar suas tarefas no sistema.

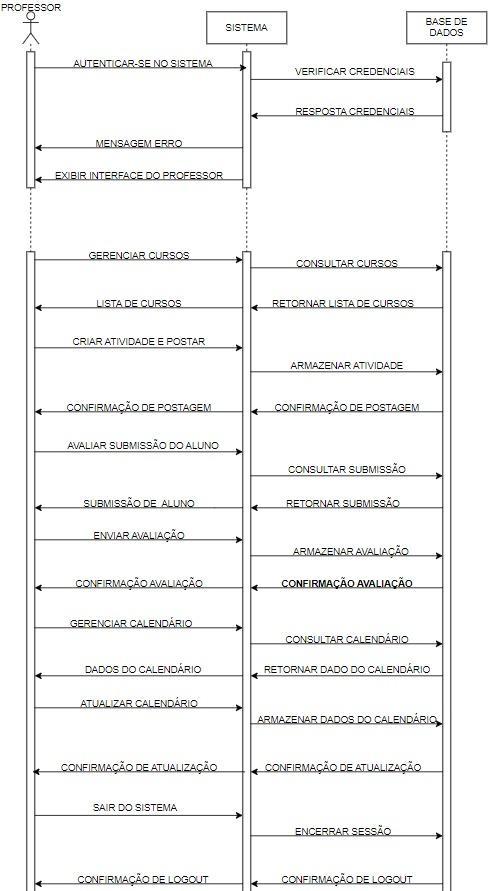


Imagem 6 - Diagrama de Sequência do Professor.

Fonte: Autor

### Considerações

**Autenticação:** A autenticação é a primeira interação, garantindo que apenas usuários autorizados possam acessar as funcionalidades do sistema.

**Seleção de Curso:** O professor seleciona o curso em que deseja trabalhar, facilitando a organização e foco nas atividades relacionadas.

**Criação e Edição de Atividades:** O professor pode criar e editar atividades, permitindo a atualização contínua dos conteúdos e tarefas.

**Interação com Alunos:** O professor pode responder a perguntas dos alunos e verificar as avaliações dessas respostas, promovendo uma comunicação eficaz.

**Verificação de Entregas:** O professor pode acompanhar as atividades entregues pelos alunos, permitindo uma avaliação contínua do progresso acadêmico.

**Log Out:** O professor encerra a sessão de forma segura, garantindo a proteção das informações do sistema.

Este diagrama de sequência fornece uma visão detalhada das interações do professor com o sistema, facilitando a compreensão do fluxo de eventos e a comunicação necessária para o desempenho eficaz das funções do professor.

### 4.3.3 Aluno

O diagrama de sequência do aluno ilustra as interações entre o aluno e o sistema para realizar diversas atividades, desde a autenticação até a visualização de materiais, verificação de notas, participação em tutoria e fórum, e a postagem de atividades. Este diagrama detalha o fluxo de eventos e a comunicação necessária para que o aluno possa executar suas tarefas no sistema.

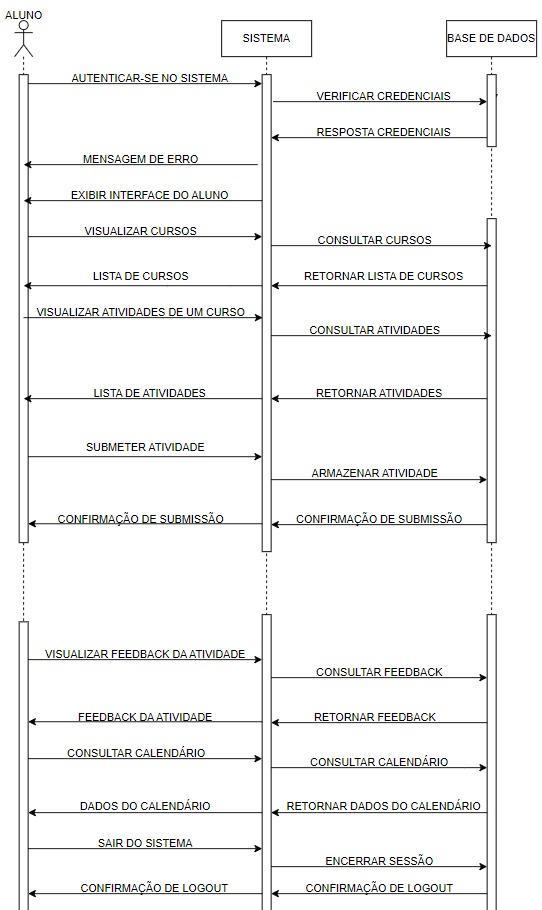


Imagem 7 - Diagrama de Sequência do Aluno.

Fonte: Autor

### Considerações

**Autenticação:** A autenticação garante que apenas alunos autorizados possam acessar as funcionalidades do sistema.

**Visualização de Materiais:** O aluno pode visualizar os materiais disponíveis para os cursos em que está matriculado.

**Seleção de Atividade:** O aluno pode selecionar e visualizar detalhes das atividades propostas.

**Verificação de Notas:** O aluno pode verificar suas notas para acompanhar seu desempenho acadêmico.

**Solicitação de Tutoria:** O aluno pode solicitar tutoria, facilitando o acompanhamento individualizado.

**Postagem de Pergunta:** O aluno pode postar perguntas para os professores, promovendo a interação e esclarecimento de dúvidas.

**Participação no Fórum:** O aluno pode participar de discussões em fóruns, colaborando com colegas para tirar dúvidas.

**Log Out:** O aluno encerra a sessão de forma segura, garantindo a proteção das informações do sistema.

Este diagrama de sequência fornece uma visão detalhada das interações do aluno com o sistema, facilitando a compreensão do fluxo de eventos e a comunicação necessária para o desempenho eficaz das funções do aluno.

# 5. Banco de Dados

## 5.1 Entidade

O modelo conceitual do banco de dados para o sistema educacional ilustra as entidades principais e suas relações, oferecendo uma visão abstrata da estrutura de dados necessária. As entidades incluem Administrador, Professor, Aluno, Turma, Calendário, Aula, Conteudo\_Assincrono, Atividades e Atividades\_Entrega. Administrador gerencia o sistema, enquanto Professor e Aluno representam os usuários principais com papéis distintos. Turma organiza alunos e professores em grupos de estudo, conectados ao Calendário para planejamento das atividades.

A tabela Aula contém detalhes das sessões de ensino, relacionando-se com Conteudo\_Assincrono para material de estudo e Atividades para as tarefas designadas aos alunos. Atividades\_Entrega monitora as submissões das atividades, assegurando o acompanhamento do desempenho dos alunos. A entidade Turma\_Aluno interliga Turma e Aluno, detalhando as matrículas e participações nas aulas. Este modelo conceitual define a estrutura básica e as interações essenciais para suportar o funcionamento do sistema educacional.

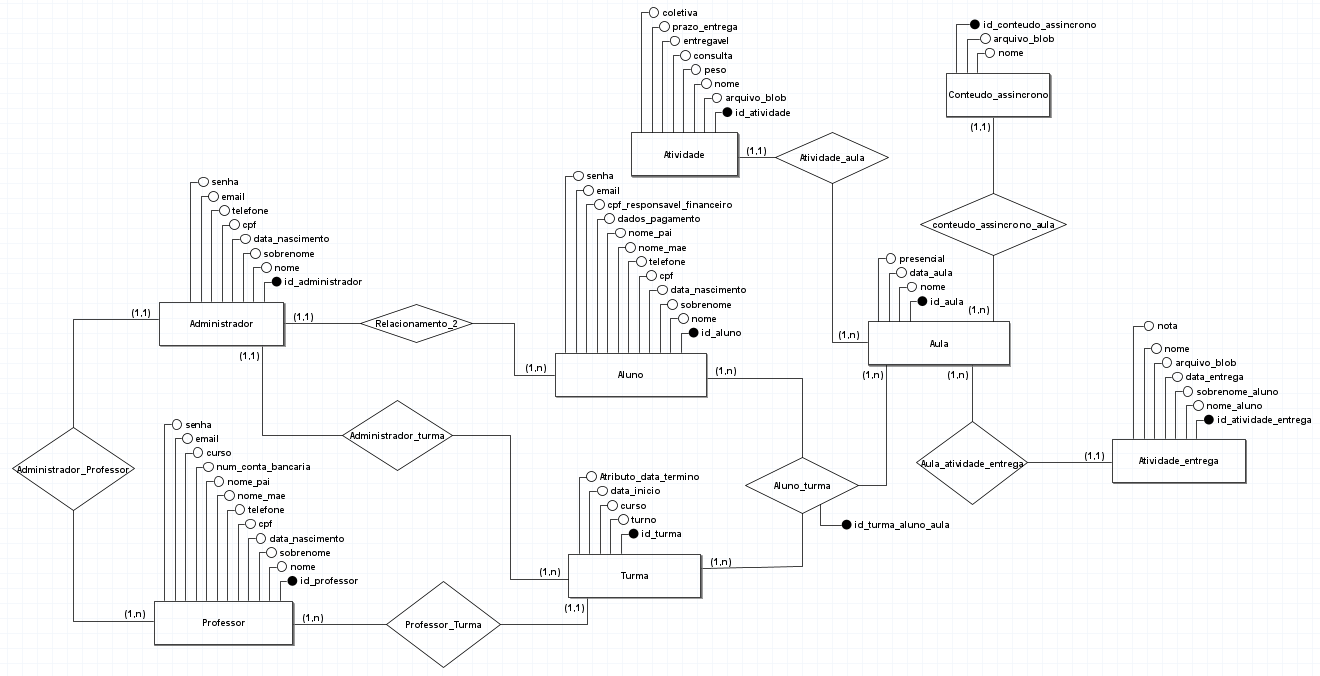


Imagem 8 - Modelo Conceitual do Banco de Dados.

Fonte: Autor

## 5.2 Dicionário de dados

O modelo lógico do banco de dados para o sistema educacional é composto por diversas tabelas inter-relacionadas que organizam as informações de usuários e atividades acadêmicas. A tabela Administrador armazena os dados dos administradores do sistema, enquanto as tabelas Professor e Aluno armazenam os dados dos respectivos usuários. A tabela Turma contém informações sobre as turmas, que são conectadas à tabela Calendário para definir o cronograma das aulas.

A tabela Turma\_Aluno registra quais alunos estão matriculados em quais turmas e se relaciona com a tabela Aula, que guarda detalhes sobre cada aula ministrada. A tabela Aula também se conecta com Conteudo\_Assincrono, armazenando os materiais de estudo disponibilizados, e com Atividades, que detalham as atividades propostas aos alunos. Finalmente, a tabela Atividades\_Entrega rastreia as submissões das atividades pelos alunos, permitindo um acompanhamento detalhado do progresso acadêmico.

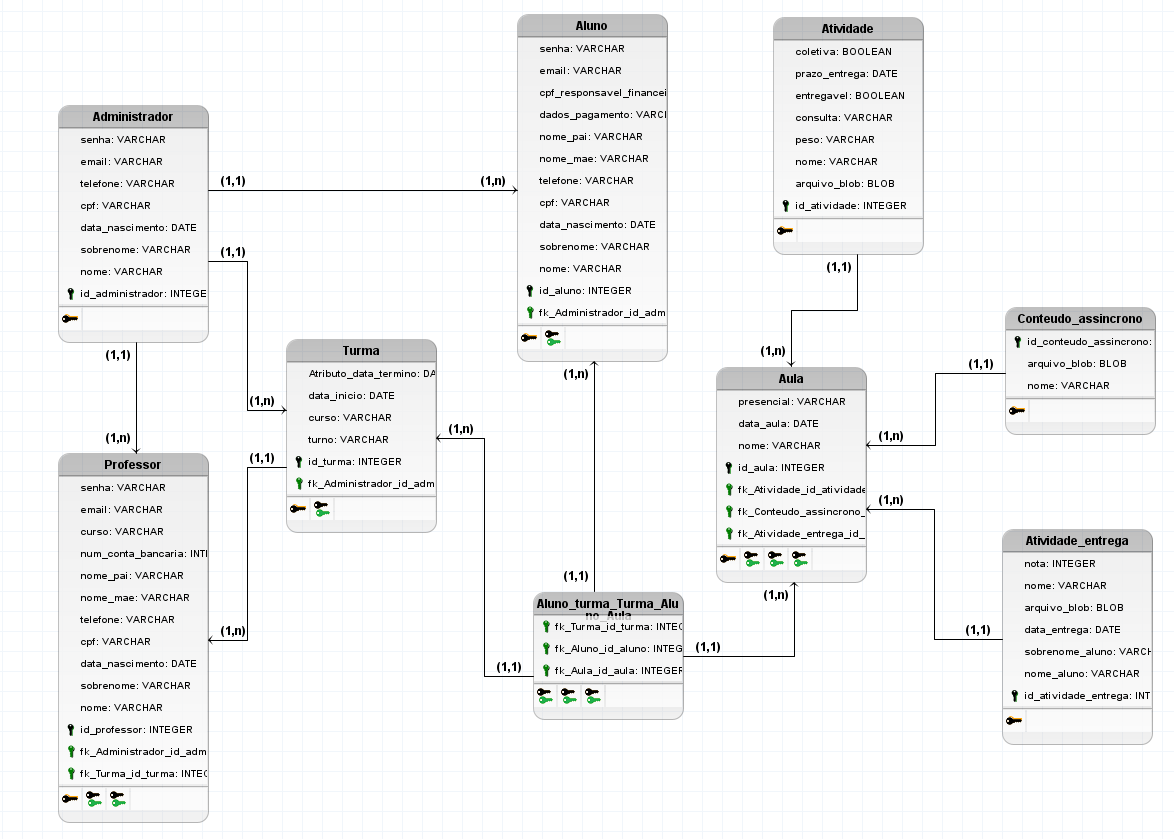


Imagem 9 - Modelo Lógico do Banco de Dados.

Fonte: Autor

## 5.3 DDL

O modelo físico do banco de dados, descrito pelo DDL (Data Definition Language), implementa a estrutura detalhada do sistema educacional em um SGBD (Sistema de Gerenciamento de Banco de Dados). Este modelo inclui a criação de tabelas e definição de suas colunas, tipos de dados, chaves primárias e estrangeiras para garantir a integridade referencial. A tabela Administrador armazena dados de administradores, enquanto Professor e Aluno contêm informações específicas dos usuários.

A tabela Turma é definida para agrupar alunos e professores, e se relaciona com o “Calendário” para definir o cronograma das aulas. A tabela Turma\_Aluno conecta Turma e Aluno, registrando matrículas. A tabela Aula inclui detalhes das sessões de ensino, relacionando-se com Conteudo\_Assincrono para materiais de estudo e Atividades para as tarefas. Atividades\_Entrega rastreia as submissões das atividades pelos alunos.

Chaves primárias são estabelecidas para identificar unicamente cada registro, enquanto chaves estrangeiras garantem as relações entre tabelas, assegurando um banco de dados relacional robusto e coerente que suporta todas as funcionalidades requeridas pelo sistema educacional.

### 5.3.1 Código do Banco

Banco de dados: `educ`

**CREATE TABLE** Administrador (

senha VARCHAR,

email VARCHAR,

telefone VARCHAR,

cpf VARCHAR,

data\_nascimento DATE,

sobrenome VARCHAR,

nome VARCHAR,

id\_administrador INTEGER PRIMARY KEY

);

**CREATE TABLE** Aluno (

senha VARCHAR,

email VARCHAR,

cpf\_responsavel\_financeiro VARCHAR,

dados\_pagamento VARCHAR,

nome\_pai VARCHAR,

nome\_mae VARCHAR,

telefone VARCHAR,

cpf VARCHAR,

data\_nascimento DATE,

sobrenome VARCHAR,

nome VARCHAR,

id\_aluno INTEGER PRIMARY KEY,

fk\_Administrador\_id\_administrador INTEGER

);

**CREATE TABLE** Turma (

Atributo\_data\_termino DATE,

data\_inicio DATE,

curso VARCHAR,

turno VARCHAR,

id\_turma INTEGER PRIMARY KEY,

fk\_Administrador\_id\_administrador INTEGER

);

**CREATE TABLE** Professor (

senha VARCHAR,

email VARCHAR,

curso VARCHAR,

num\_conta\_bancaria INTEGER,

nome\_pai VARCHAR,

nome\_mae VARCHAR,

telefone VARCHAR,

cpf VARCHAR,

data\_nascimento DATE,

sobrenome VARCHAR,

nome VARCHAR,

id\_professor INTEGER PRIMARY KEY,

fk\_Administrador\_id\_administrador INTEGER,

fk\_Turma\_id\_turma INTEGER

);

**CREATE TABLE** Aula (

presencial VARCHAR,

data\_aula DATE,

nome VARCHAR,

id\_aula INTEGER PRIMARY KEY,

fk\_Atividade\_id\_atividade INTEGER,

fk\_Conteudo\_assincrono\_id\_conteudo\_assincrono INTEGER,

fk\_Atividade\_entrega\_id\_atividade\_entrega INTEGER

);

**CREATE TABLE** Atividade (

coletiva BOOLEAN,

prazo\_entrega DATE,

entregavel BOOLEAN,

consulta VARCHAR,

peso VARCHAR,

nome VARCHAR,

arquivo\_blob BLOB,

id\_atividade INTEGER PRIMARY KEY

);

**CREATE TABLE** Conteudo\_assincrono (

id\_conteudo\_assincrono INTEGER PRIMARY KEY,

arquivo\_blob BLOB,

nome VARCHAR

);

**CREATE TABLE** Atividade\_entrega (

nota INTEGER,

nome VARCHAR,

arquivo\_blob BLOB,

data\_entrega DATE,

sobrenome\_aluno VARCHAR,

nome\_aluno VARCHAR,

id\_atividade\_entrega INTEGER PRIMARY KEY

);

**CREATE TABLE** Aluno\_turma\_Turma\_Aluno\_Aula (

fk\_Turma\_id\_turma INTEGER,

fk\_Aluno\_id\_aluno INTEGER,

fk\_Aula\_id\_aula INTEGER

);

# 6. Telas do Projeto e suas Definições

Chegamos na parte da montagem das telas, conversamos então com uma pessoa do design e pedimos ajuda para que ela montasse algumas telas para que pudéssemos então ter uma visão e também uma ideia melhor de como as telas do nosso AVA ficariam, após vermos alguns dos modelos e das ideias começamos então a montar o front de nosso site, a primeira parte foi a tela de login.

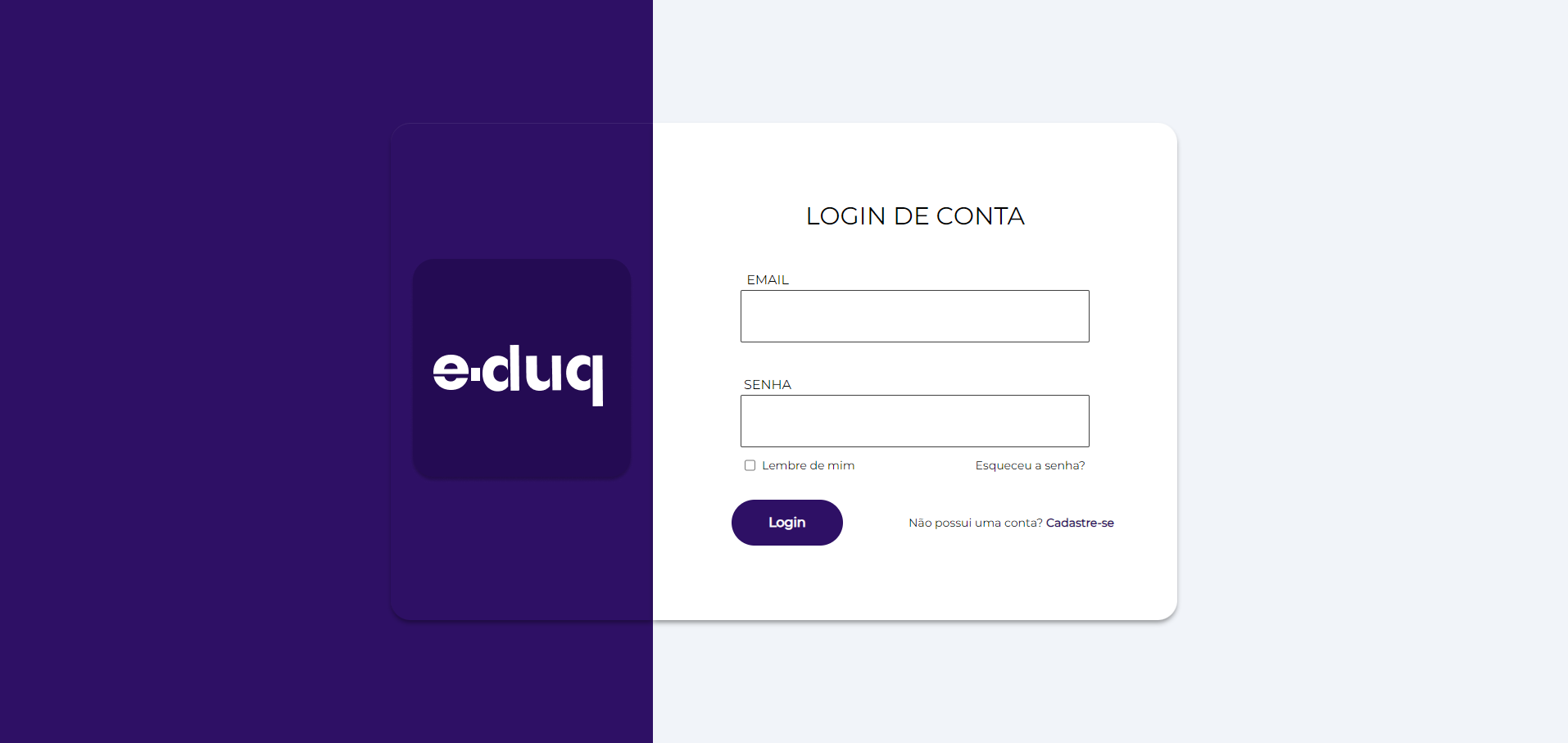


Imagem 11 - Tela de login do Sistema.

Fonte: Autor

Nela tem o Logo do Eduq no lado esquerdo e ao lado direito os campos para e-mail e senha, nele também contém uma opção para caso o usuário tenha esquecido sua senha, a opção lembre de mim e o botão para fazer o login, tem também uma opção para cadastro que foi colocado devido a necessidade de cadastrar os usuários e fazer os testes necessários de funcionamento do sistema.

Com o front da página de login pronto, demos início ao “Back” da mesma, para que os dados cadastrados dos administradores, professores e alunos fossem buscados e permitissem o acesso à página inicial de cada um, sendo a do administrador a página a qual tem acesso aos seus respectivos papéis, o professor tendo acesso a página onde terá acesso a turmas as quais ele dá aulas, e a página dos alunos que terão acesso aos conteúdos das aulas, as atividades, provas, notas dentre outras opções das quais estarão disponíveis para ele.

Em seguida demos início a página inicial do administrador, nela estão presentes todos os dados necessários para que o administrador tenha fácil acesso a tudo o que precisar.

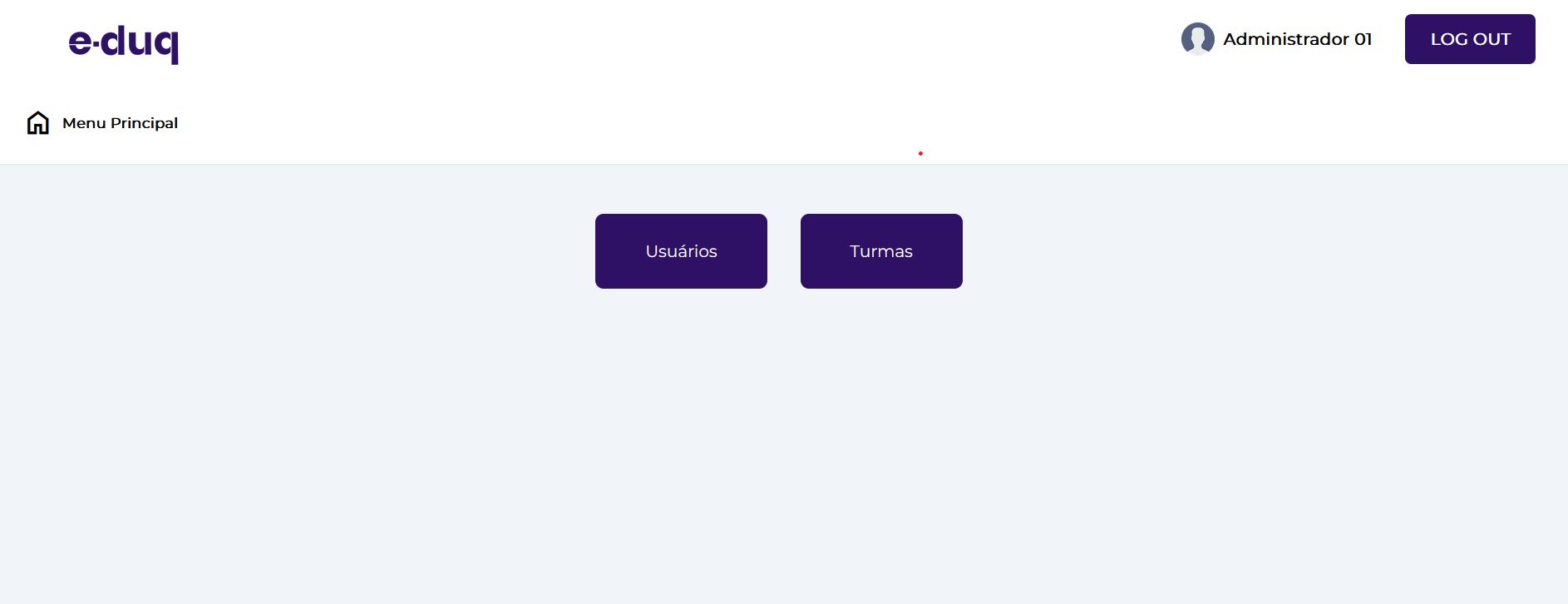


Imagem 12 - Tela Inicial do Administrador no Sistema.

Fonte: Autor

O front da página do administrador conta com o Header onde fica o logo do “Eduq” na parte superior esquerda e logo abaixo dele a opção de “Menu Principal” e no lado superior direito fica o nome cadastrado pelo administrador e o botão de “Logout”, na página de administrador tem os botões com as opções de “Usuários”, o que inclui todos os usuários cadastrados e “Turmas”, que contém todas as turmas cadastradas no sistema.

O “Back” do mesmo foi desenvolvido em PHP e nele estão incluídos o gerenciar\_turmas e gerenciar\_usuarios, as partes de cadastrar aluno e cadastrar professor já existem, mas ainda não foram implementadas completamente na página no administrador e por isso as opções ainda não estão disponíveis para acesso direto pela página inicial dele.

O nível professor o qual terá acesso ao que lhe for atribuído pelo administrador, como as turmas em que ele irá dar aula, dentro delas o professor poderá mexer nos conteúdos da aula, na lista de chamadas, criar provas e atividades dentre várias outras opções as quais ainda estão sendo implementadas e várias outras que ainda serão adicionadas.

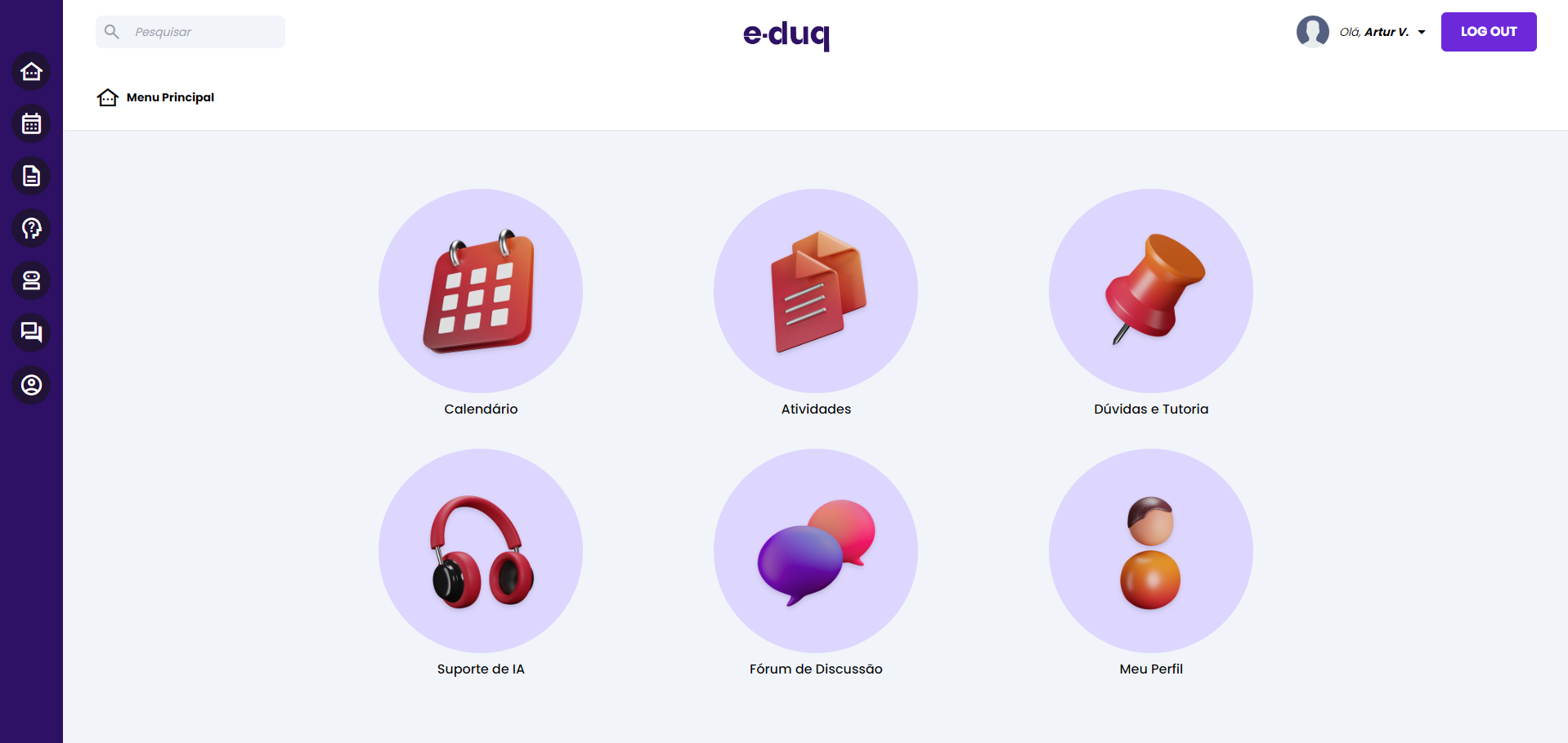


Imagem 13 - Tela Inicial do Professor no Sistema.

Fonte: Autor

O front da página do professor contém o Header, mas diferente da página do administrador no canto esquerdo da página se encontra uma barra de pesquisa onde o professor pode pesquisar algo para acesso rápido, abaixo tem a opção do “Menu Principal”, no centro do header se encontra o logo do “Eduq” e do lado direito se encontra o nome do usuário e um botão de “Logout”. Na página inicial do professor se encontra o Calendário, Atividades, Dúvidas e Tutoria, Suporte de IA, Fórum de Discussão e Meu Perfil, todas as opções em botões bem definidos os quais redirecionam para suas respectivas funções. No lado esquerdo da página contém uma aba com versões miniaturas das mesmas funções, tudo foi desenvolvido de tal modo para que usuários das mais variadas idades pudessem acessar o sistema e entender de modo rápido e sem problemas.

O “Back” da parte do professor foi desenvolvido também em PHP e várias das opções já estão disponíveis no sistema, como: criarAtividade, criarAula, criarConteudo, editarAtividade, editarConteudo, professor\_perfil e turma, cada uma destas opções já foram atribuídas no sistema e apesar de algumas ainda não estarem completamente implementadas na página do professor elas já estão funcionais e prontas para o uso.

O nível aluno no sistema é projetado para oferecer acesso fácil e direto às funcionalidades essenciais para a gestão de suas atividades acadêmicas. Assim como o professor, o aluno tem acesso a uma interface intuitiva

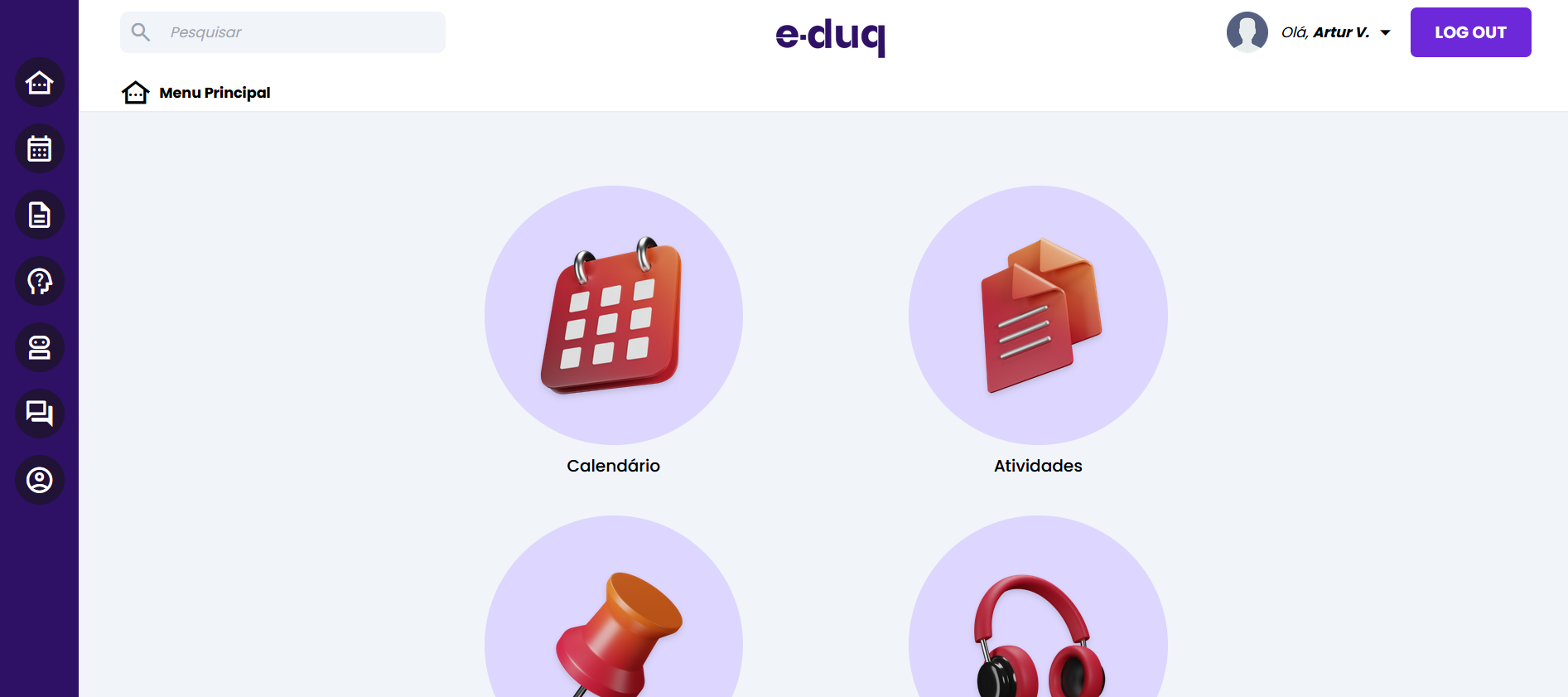


Imagem 15 - Tela Inicial do Aluno no Sistema.

Fonte: Autor

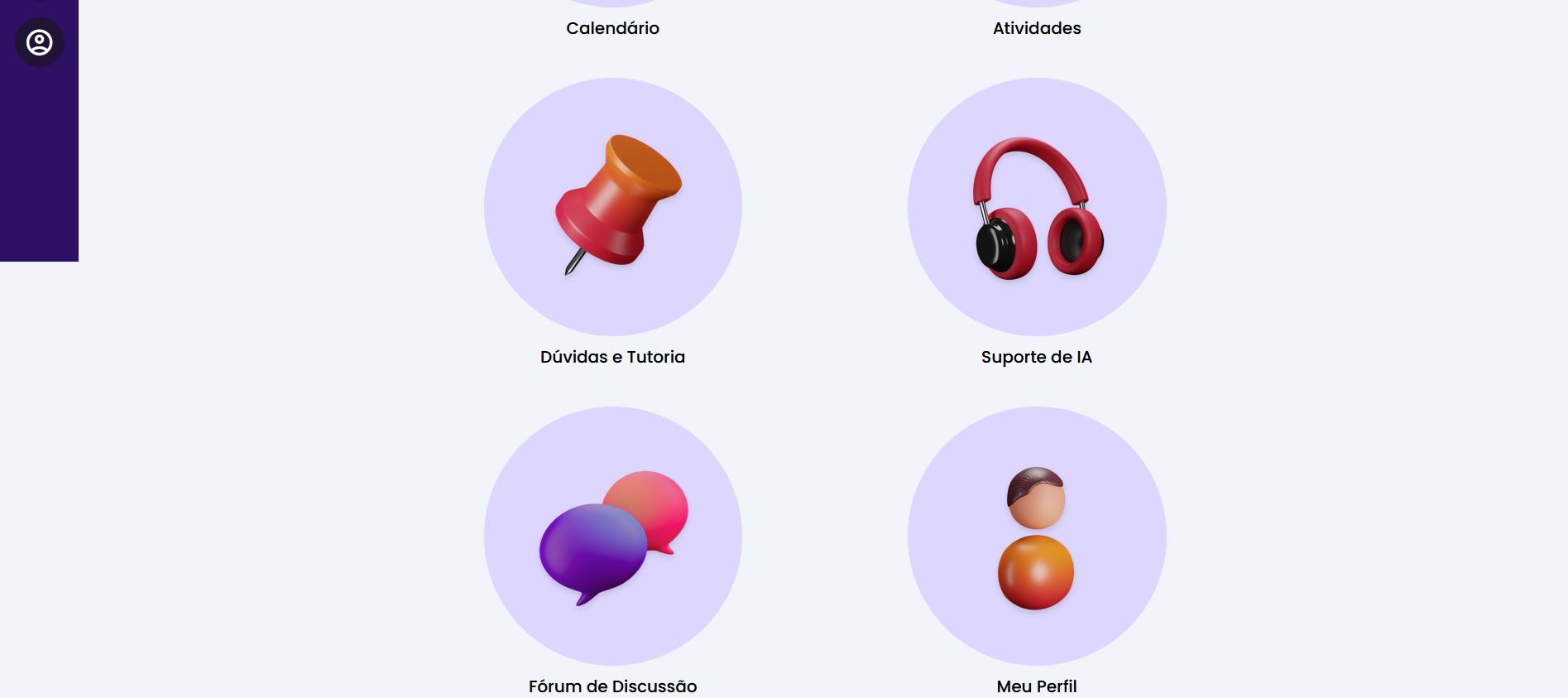


Imagem 16 - Tela Inicial do Aluno no Sistema.

Fonte: Autor

Na página inicial do aluno, há botões bem definidos que redirecionam para várias funções essenciais: Calendário, Atividades, Dúvidas e Tutoria, Suporte de IA, Fórum de Discussão e Meu Perfil. Cada botão leva o aluno diretamente à área correspondente, facilitando o acesso a todas as funcionalidades necessárias para a gestão de seus estudos e interações.

Além disso, no lado esquerdo da página, uma aba contém miniaturas das mesmas funções principais, proporcionando uma navegação rápida e conveniente. Esta organização visual e funcional foi desenvolvida para garantir que todos os alunos, independentemente da idade ou familiaridade com a tecnologia, possam utilizar o sistema de maneira eficiente e sem dificuldades.

O “Back-end” da parte do aluno no sistema também foi desenvolvido em PHP, assegurando a eficiência e a funcionalidade necessárias para a gestão das atividades acadêmicas. Várias das opções já estão disponíveis no sistema, como: alun\_atividade\_entrega, alun\_atividade e alun\_conteudo\_assincrono. Essas funcionalidades permitem que os alunos entreguem atividades, acessem suas tarefas e visualizem conteúdos assíncronos de forma prática e organizada. Embora algumas dessas opções ainda não estejam completamente implementadas na interface do aluno, elas já estão operacionais e prontas para uso, garantindo que o sistema continue a oferecer suporte eficaz às necessidades dos estudantes.

# 7. Engenharia de Testes

## 7.1 Teste de Conexão com o Banco de Dados:

### **7.1.1 Configuração do Ambiente**

Antes de iniciar os testes de conexão, foi essencial garantir que o ambiente de desenvolvimento estivesse corretamente configurado. Isso incluiu verificar se o servidor MySQL estava em execução e se as credenciais de acesso fornecidas no script PHP estavam corretas. Qualquer discrepância nessa configuração poderia levar a falhas nos testes de conexão.

### **7.1.2 Script de Teste**

Um script PHP foi desenvolvido para tentar estabelecer uma conexão com o banco de dados utilizando a função mysqli\_connect(). Esse script foi projetado para lidar com possíveis erros de conexão, como credenciais inválidas ou indisponibilidade do servidor. Durante o desenvolvimento do script, foram incluídos logs detalhados para facilitar a depuração de problemas de conexão.

### **7.1.3 Execução do Teste**

O script foi executado em diferentes momentos do desenvolvimento para verificar se a conexão com o banco de dados era bem-sucedida. Mensagens de feedback foram exibidas para indicar se a conexão foi estabelecida com êxito ou se ocorreu algum erro. Testes adicionais foram realizados para simular condições de erro, como credenciais incorretas ou servidor indisponível, garantindo que o sistema fosse robusto o suficiente para lidar com essas situações.

**Considerações**

Para garantir que o sistema pudesse se conectar de forma eficiente e confiável ao banco de dados, foram realizados testes abrangentes de conexão. Utilizando ferramentas como PhpMyAdmin e MySQL Workbench, verificamos a configuração das credenciais de acesso, garantindo que estivessem corretas e que o sistema pudesse se conectar sem problemas ao banco de dados. Além disso, foram realizados testes para verificar a disponibilidade do banco de dados e a capacidade de executar consultas SQL com sucesso.

## 7.2 Teste de Integridade do Banco de Dados:

### **7.2.1 Verificação das Chaves Estrangeiras**

Consultas SQL foram executadas para verificar se as chaves estrangeiras estavam corretamente aplicadas nas tabelas relacionadas. Isso incluiu a verificação de integridade referencial para garantir que cada valor em uma coluna de chave estrangeira correspondesse a um valor válido na tabela relacionada. Quaisquer violações de integridade foram identificadas e corrigidas para garantir a consistência dos dados.

### **7.2.2 Validação dos Dados**

Uma série de consultas SQL foi desenvolvida para validar os dados armazenados nas tabelas do banco de dados. Isso envolveu a verificação de campos obrigatórios para garantir que estivessem devidamente preenchidos, bem como a validação de tipos de dados e intervalos permitidos. Quaisquer inconsistências ou dados inválidos foram corrigidos de acordo com as regras de negócio definidas.

### **7.2.3 Teste de CRUD**

As operações de criação, leitura, atualização e exclusão (CRUD) foram executadas em diferentes tabelas do banco de dados para garantir que o sistema estivesse funcionando conforme o esperado. Cada operação foi cuidadosamente verificada para garantir que não causasse inconsistências nos dados e que os resultados fossem consistentes com as expectativas. Esses testes foram essenciais para garantir a integridade dos dados em todas as etapas do ciclo de vida do sistema.

**Considerações**

Os testes de integridade dos dados foram realizados para garantir que as informações armazenadas no banco de dados estivessem precisas, consistentes e livres de erros. Utilizando ferramentas como MySQL Workbench, examinamos o esquema do banco de dados e as relações entre as tabelas para verificar se estavam corretamente definidas. Além disso, foram realizados testes para validar a integridade referencial, garantindo que as chaves estrangeiras estivessem configuradas corretamente e que não houvesse violações de integridade nos dados.

## 7.3 Testes de Frontend

### 7.3.1 Teste de Interface do Usuário

# Abrimos a aplicação em diferentes navegadores, como Google Chrome, Firefox e Microsoft Edge, em diferentes dispositivos, incluindo desktops, laptops, tablets e smartphones, para verificar o layout, estilo e funcionalidades. Durante esse teste, verificamos se todos os elementos da interface do usuário estavam renderizando corretamente em cada navegador e dispositivo, garantindo uma experiência consistente para os usuários, independentemente do dispositivo ou navegador que estivessem usando.

### **7.3.2 Teste de Responsividade:**

Utilizamos as ferramentas de desenvolvimento integradas dos navegadores, como as opções de inspeção do Google Chrome e Firefox, para simular diferentes tamanhos de tela e verificar se a interface do usuário se ajustava corretamente. Testamos diferentes breakpoints definidos no código CSS para garantir que os elementos da interface do usuário respondessem adequadamente a diferentes tamanhos de tela, mantendo a usabilidade e a legibilidade. Esses testes de responsividade foram cruciais para garantir uma experiência de usuário consistente e otimizada em uma variedade de dispositivos e tamanhos de tela.

**Considerações**

Considerando os testes de interface de usuário e de responsividade, foi possível garantir a qualidade da experiência do usuário em diferentes cenários de uso. A combinação desses testes permitiu verificar não apenas a aparência visual da aplicação, mas também sua adaptabilidade e usabilidade em dispositivos de diversos tamanhos e formatos de tela. Ao realizar esses testes, pudemos assegurar que a aplicação oferecesse uma experiência consistente e intuitiva, independentemente do dispositivo utilizado pelo usuário. O uso de ferramentas como Firefox e VS Code facilitou a execução dos testes e contribuiu para uma análise detalhada e abrangente da interface e da responsividade da aplicação.

## 7.4 Testes de Backend

### 7.4.1 Teste de Rotas e Requisições HTTP

Nesta etapa, foram realizados testes para verificar a correta configuração e funcionamento das rotas e requisições HTTP do sistema. Isso envolveu o teste de todas as rotas definidas na API, garantindo que cada uma estivesse acessível e respondesse corretamente às solicitações. Além disso, as requisições HTTP foram testadas para garantir que os métodos GET, POST, PUT e DELETE estivessem sendo utilizados de acordo com as especificações e que os dados fossem transmitidos de forma segura entre o “frontend” e o “backend”.

**Considerações**

O teste de rotas e requisições HTTP foi essencial para garantir a correta comunicação entre o “frontend” e o “backend”, assegurando que todas as funcionalidades estivessem acessíveis e funcionassem conforme o esperado. A verificação dos métodos HTTP permitiu garantir que as operações de CRUD estivessem sendo executadas adequadamente, proporcionando uma interação eficiente e segura com o banco de dados.

### 7.4.2 Teste de Lógica de Negócios

Nesta fase, os testes foram focados na validação da lógica de negócios implementada no “backend” do sistema. Isso envolveu a verificação de todas as regras de negócio definidas, garantindo que as operações realizadas pelo sistema estivessem de acordo com as expectativas e requisitos do usuário. Foram realizados testes de casos de uso para cada funcionalidade, verificando se o sistema estava respondendo corretamente às diferentes entradas e condições, e produzindo os resultados esperados.

**Considerações**

O teste de lógica de negócios foi fundamental para garantir que o sistema estivesse operando de acordo com as especificações e requisitos do usuário, assegurando a consistência e confiabilidade das operações realizadas. A validação dos casos de uso permitiu identificar e corrigir eventuais falhas na implementação da lógica de negócios, garantindo a entrega de um sistema funcional e livre de erros.

### 7.4.3 Teste de Integração

Durante esta etapa, os testes foram direcionados para verificar a integração entre os diferentes componentes do sistema. Isso incluiu a validação da interação entre o “backend” e o banco de dados, bem como a comunicação entre os diferentes módulos e serviços do sistema. Foram realizados testes para garantir que os componentes estivessem se comunicando corretamente e que as informações estivessem sendo transmitidas de forma eficiente e precisa.

**Considerações**

O teste de integração foi crucial para garantir a interoperabilidade e eficácia do sistema como um todo, assegurando que todos os componentes estivessem trabalhando em conjunto de forma harmoniosa e sem problemas. A verificação da comunicação entre os diferentes módulos e serviços permitiu identificar e corrigir eventuais falhas de integração, garantindo a entrega de um sistema coeso e funcional.

### 7.4.4 Teste de Desempenho

Nesta etapa, foram realizados testes para avaliar o desempenho e a escalabilidade do sistema. Isso incluiu a análise do tempo de resposta das diferentes operações, a capacidade de lidar com cargas de trabalho variáveis e a identificação de possíveis gargalos de desempenho. Foram utilizadas ferramentas de teste de carga para simular condições de uso reais e verificar como o sistema se comporta sob diferentes cenários.

**Considerações**

O teste de desempenho foi essencial para garantir que o sistema pudesse lidar com o volume de tráfego esperado e proporcionar uma experiência de usuário satisfatória mesmo em momentos de pico de utilização. A identificação de possíveis gargalos de desempenho permitiu otimizar o sistema e garantir uma resposta rápida e eficiente às solicitações dos usuários.

# 8. Conclusão

A documentação apresentada representa um esforço abrangente e meticuloso na concepção, desenvolvimento e teste de um sistema educacional eficaz. Desde o modelo conceitual do banco de dados até os testes de frontend e backend, cada aspecto foi cuidadosamente elaborado para fornecer uma visão detalhada e abrangente do sistema. Através dessa documentação, buscamos não apenas oferecer um guia para o desenvolvimento e implementação do sistema, mas também destacar a importância da engenharia de testes em todas as fases do projeto.

Com uma estrutura clara e informações detalhadas, essa documentação serve como um recurso valioso para garantir a integridade, desempenho e eficácia do sistema educacional proposto. Além disso, ela oferece uma base sólida para o desenvolvimento futuro e o aprimoramento contínuo do sistema, promovendo uma experiência de aprendizado intuitiva e eficaz para os usuários finais.

Além disso, ao fornecer uma visão abrangente do sistema educacional em questão, a documentação permite uma comunicação clara e eficaz entre todas as partes envolvidas no projeto. Isso inclui desenvolvedores, testadores, gerentes de projeto e clientes, garantindo que todos tenham uma compreensão comum dos requisitos, funcionalidades e objetivos do sistema. Dessa forma, a documentação não apenas serve como um guia para o desenvolvimento técnico, mas também como uma ferramenta de alinhamento e colaboração entre as diferentes equipes e stakeholders.

Considerando a complexidade e o escopo abrangente do projeto, o investimento de R$8000,00 na criação da documentação representa um valor justo e estratégico para o cliente. Esta documentação não apenas proporciona uma base sólida para o desenvolvimento do sistema educacional, mas também ajuda a mitigar riscos, acelerar o processo de desenvolvimento e garantir a entrega de um produto final de alta qualidade e desempenho. Portanto, trata-se não apenas de um custo, mas sim de um investimento essencial para o sucesso e a eficácia do projeto como um todo.

Além disso, a documentação detalhada e abrangente também serve como um recurso valioso para treinamento e capacitação de novos membros da equipe, garantindo uma transição suave e eficiente durante o ciclo de vida do projeto.

# 9. Referências

W3C. (2018). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. Retirado de:<https://www.w3.org/TR/html5/>

Tailwind Labs. (2023). *Tailwind CSS Documentation*. Retirado de:<https://tailwindcss.com/docs>

jQuery Foundation. (2023). *jQuery Documentation*. Retirado de: <https://api.jquery.com/>

The PHP Group. (2023). *PHP Manual*. Retirado de:<https://www.php.net/manual/en/>

Oracle Corporation. (2023). *MySQL Documentation*. Retirado de:<https://dev.mysql.com/doc/>

# 10. Anexos

## 10. 1 Telas dos códigos

### 10.1.1 Administrador

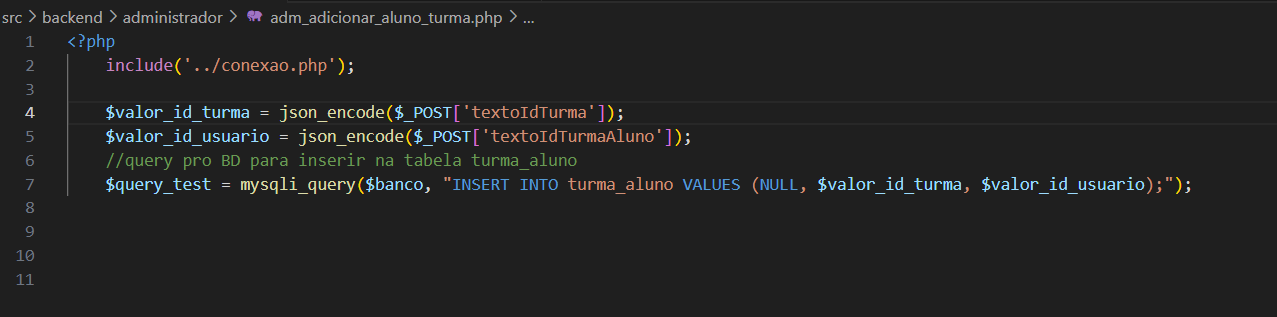


Imagem 17 - Código da Parte Adicionar.

Fonte: Autor

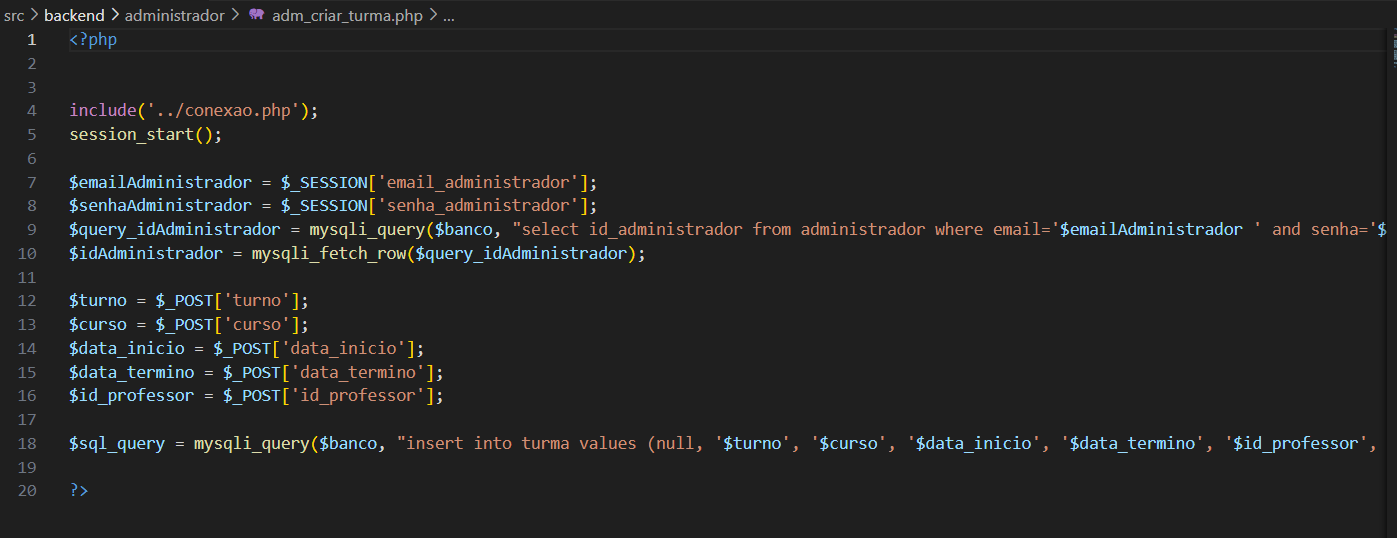


Imagem 18 - Código da Parte Criar.

Fonte: Autor



Imagem 19 - Código da Parte Editar Dados.

Fonte: Autor



Imagem 20 - Código da Parte Listar Dados.

Fonte: Autor

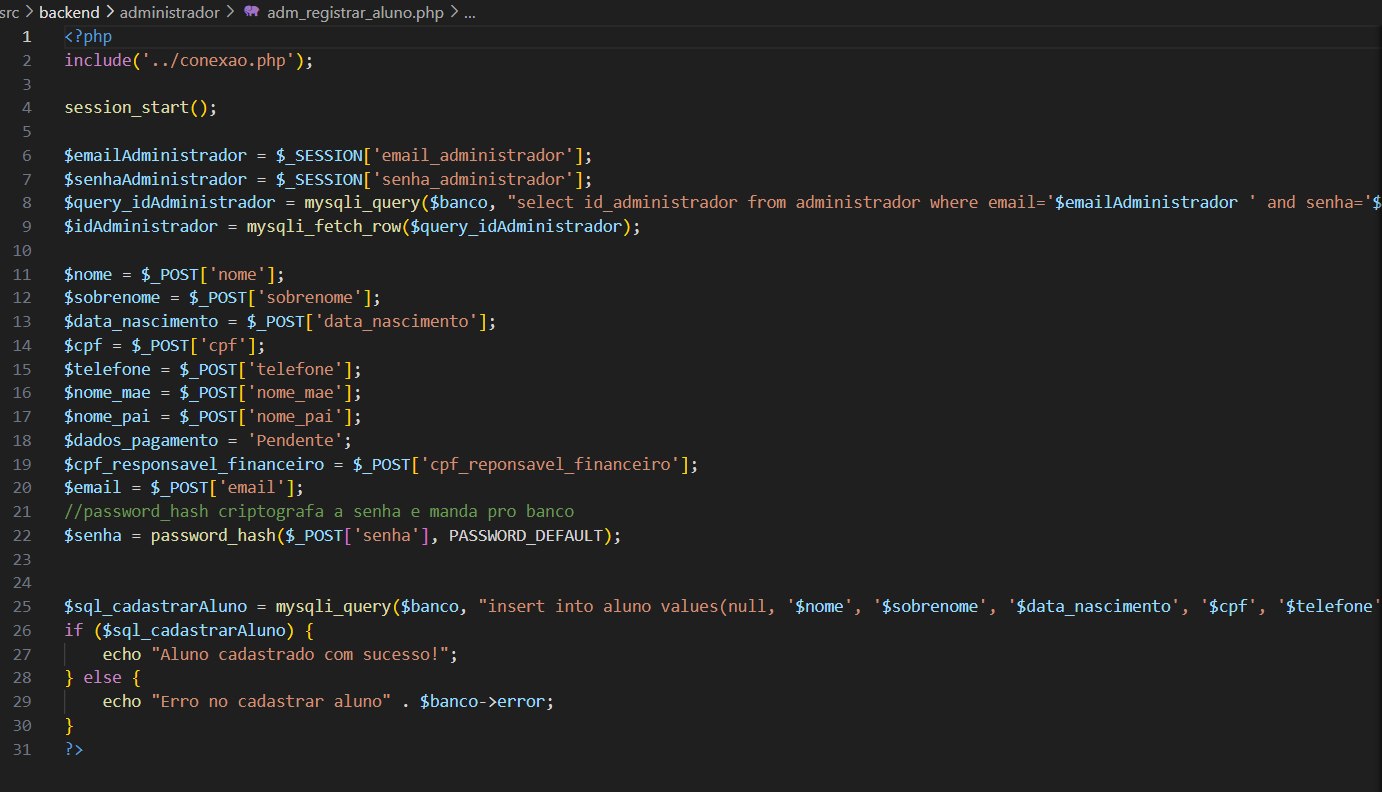


Imagem 21 - Código da Parte Registrar Alunos.

Fonte: Autor

### 10.1.2 Descrição de algumas funções no código do Administrador

adm\_adicionar\_… - Código responsável por cadastrar um aluno em uma turma

adm\_criar… - Código responsável por criar uma turma e já adicionar um professor cadastrado na turma.

adm\_editar… - Código responsável para que o usuário consiga alterar informações de turmas e usuários.

adm\_listar… - Código responsável por listar as turmas e os usuários para o usuário.

adm\_registrar… - Código responsável por cadastrar turmas e usuários.

## 10.2 Professor

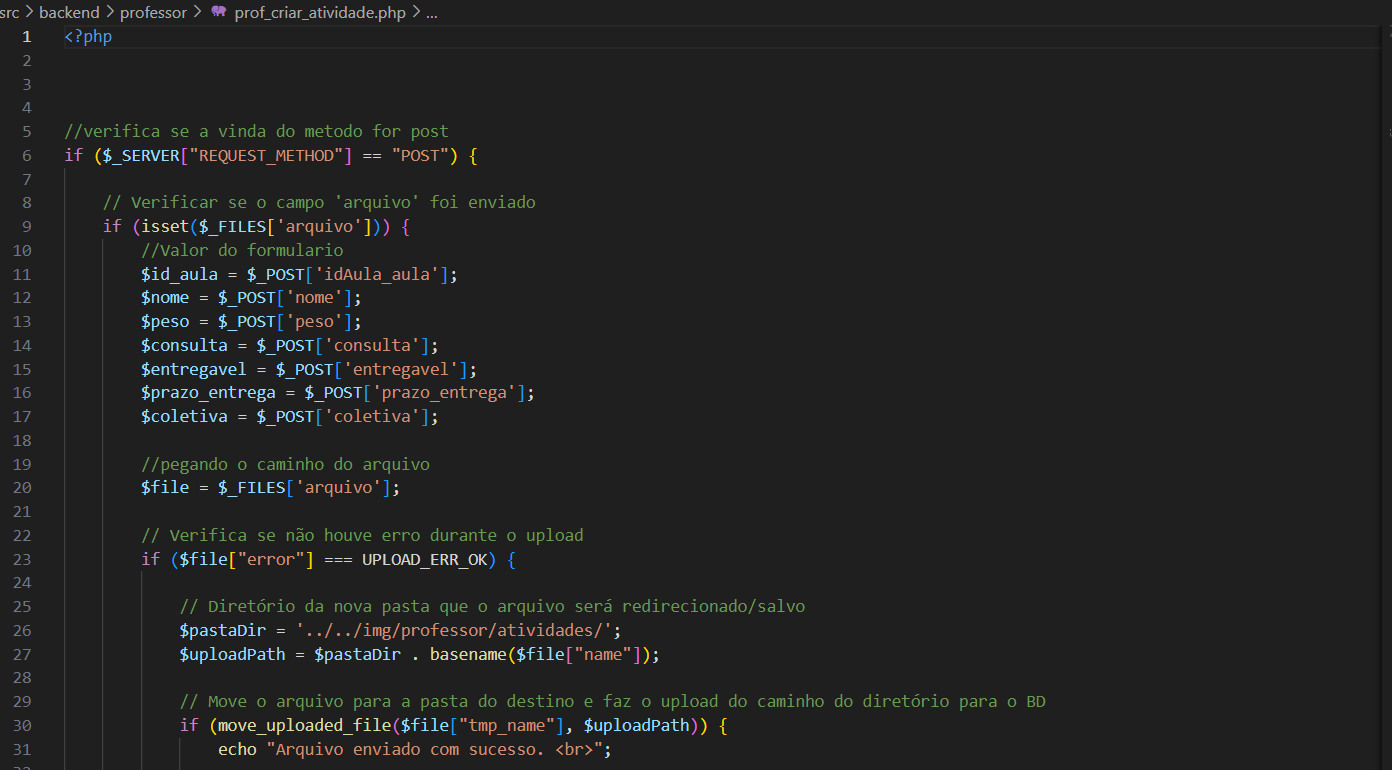
****

Imagem 22 - Código da Parte Criar Atividade.

Fonte: Autor

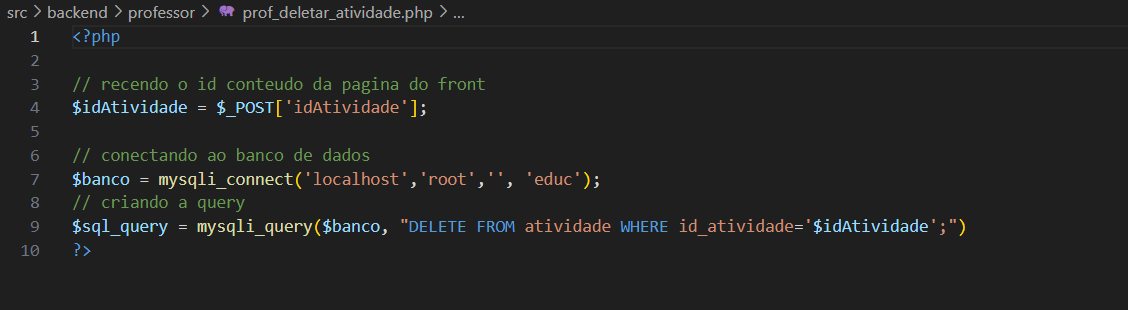


Imagem 23 - Código da Parte Deletar Atividade.

Fonte: Autor

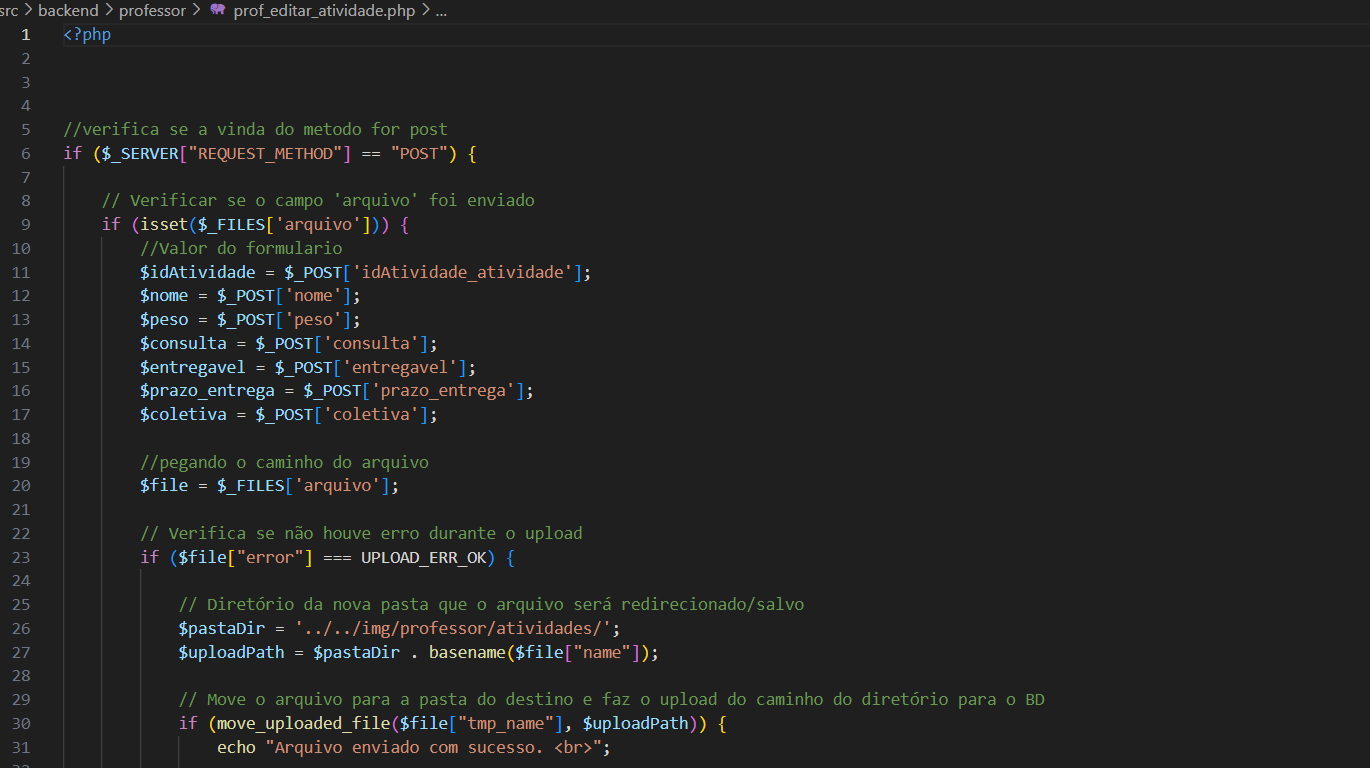
****

Imagem 24 - Código da Parte Editar Atividade.

Fonte: Autor

### 10.2.1 Descrição de algumas funções no código do Professor

prof\_criar\_… - Código responsável por criar aula, atividade e conteúdo.

prof\_deletar\_… - Código responsável por deletar atividades e conteúdo.

prof\_editar\_… - Código responsável para que o usuário consiga alterar as informações de atividades e conteúdo.

prof\_atividade\_entrega - Código responsável por listar as atividades entregues pelo aluno

prof\_aula - Código responsável por listar as atividades e os conteúdos dos alunos

prof\_perfil - Código responsável por listar as turmas e os dados do perfil do usuário

prof\_turma - Código responsável por listar as aulas da turma

## 10.3 Aluno

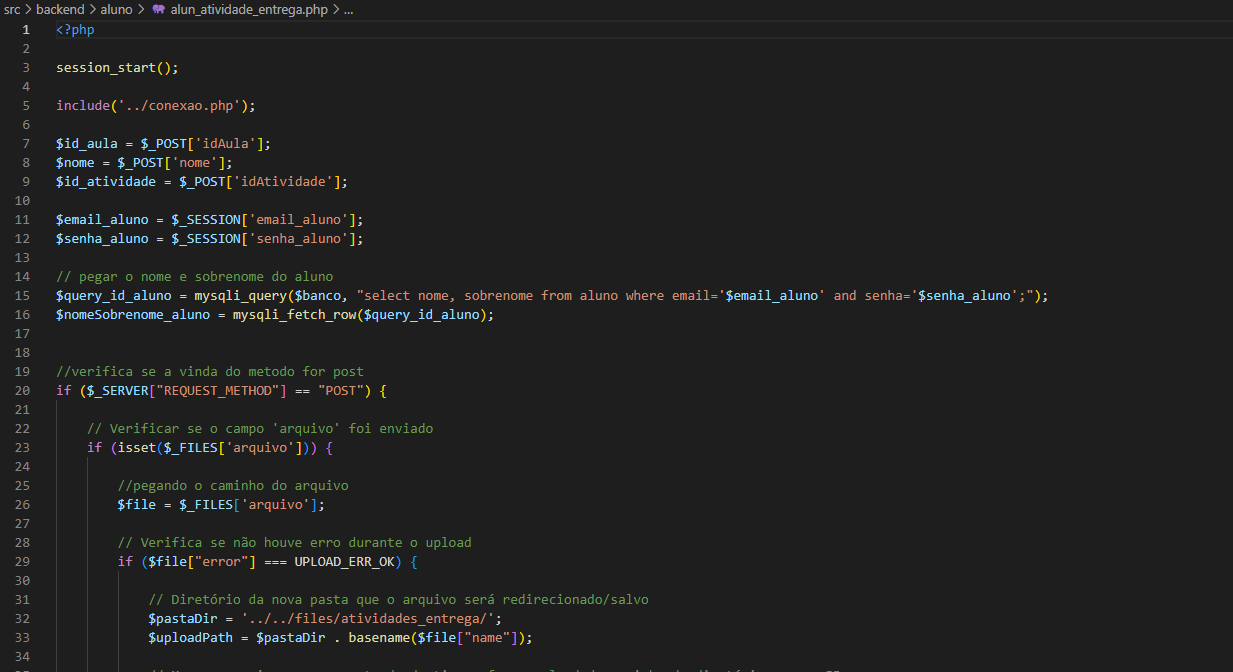


Imagem 25 - Código da Parte de Entrega de Atividade.

Fonte: Autor

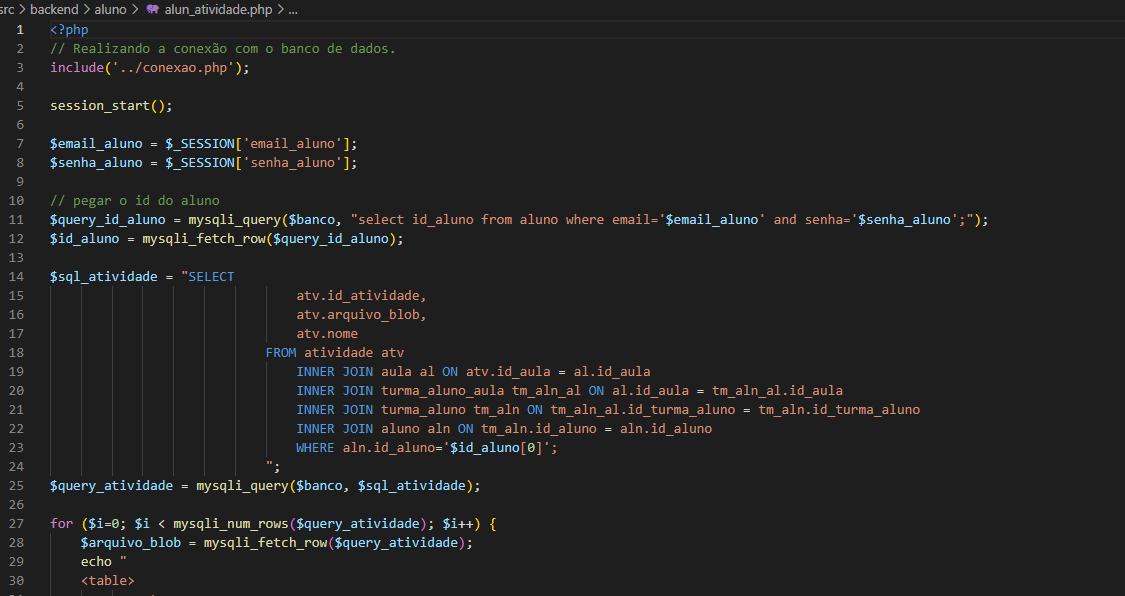


Imagem 26 - Código da Parte Atividade.

Fonte: Autor

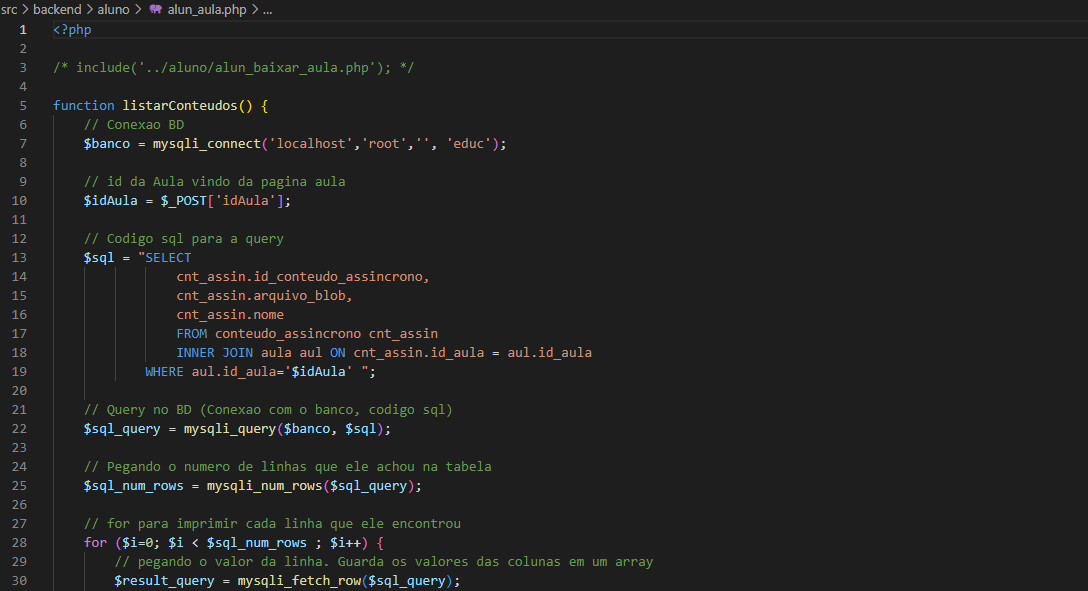


Imagem 27 - Código da Parte Aula.

Fonte: Autor

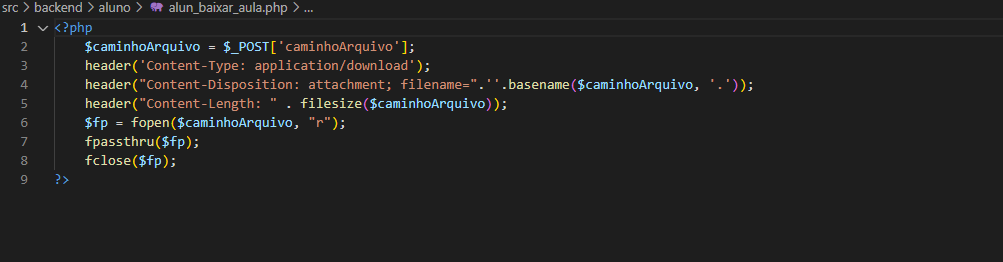


Imagem 28 - Código da Parte Baixar Aula.

Fonte: Autor

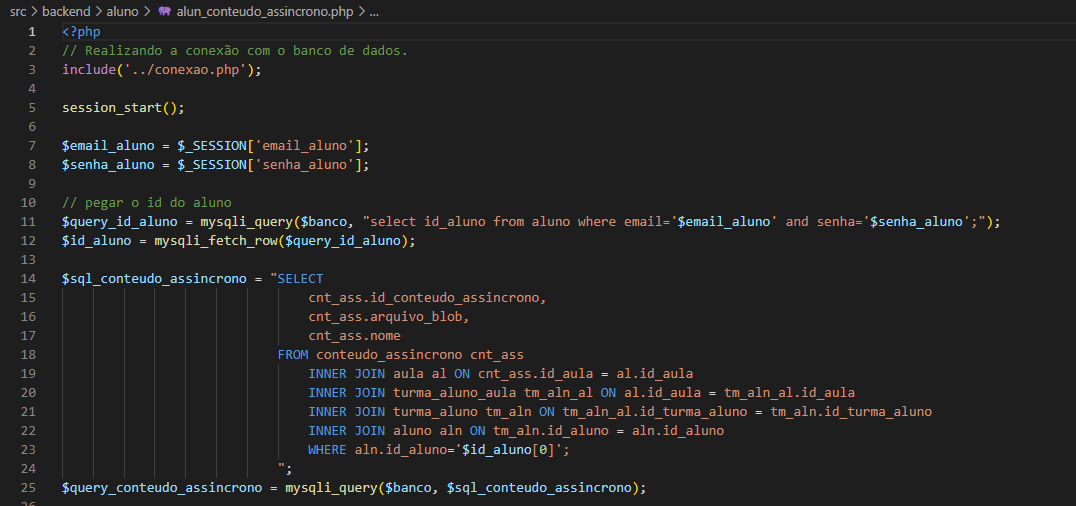


Imagem 29 - Código da Parte Conteúdo Assíncrono.

Fonte: Autor

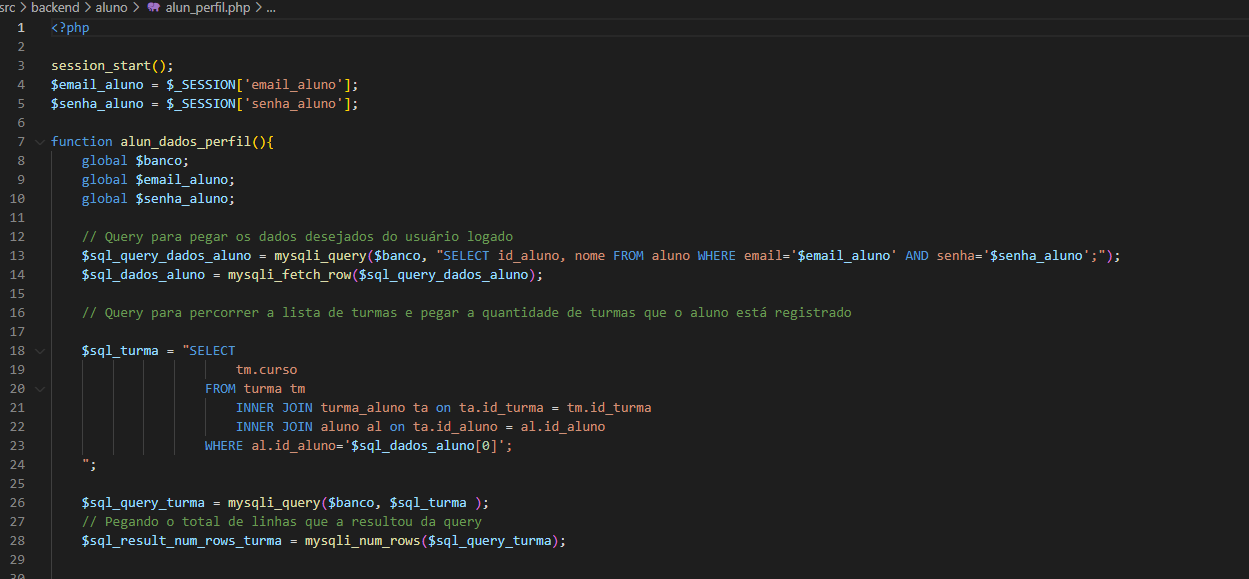


Imagem 30 - Código da Parte do Perfil.

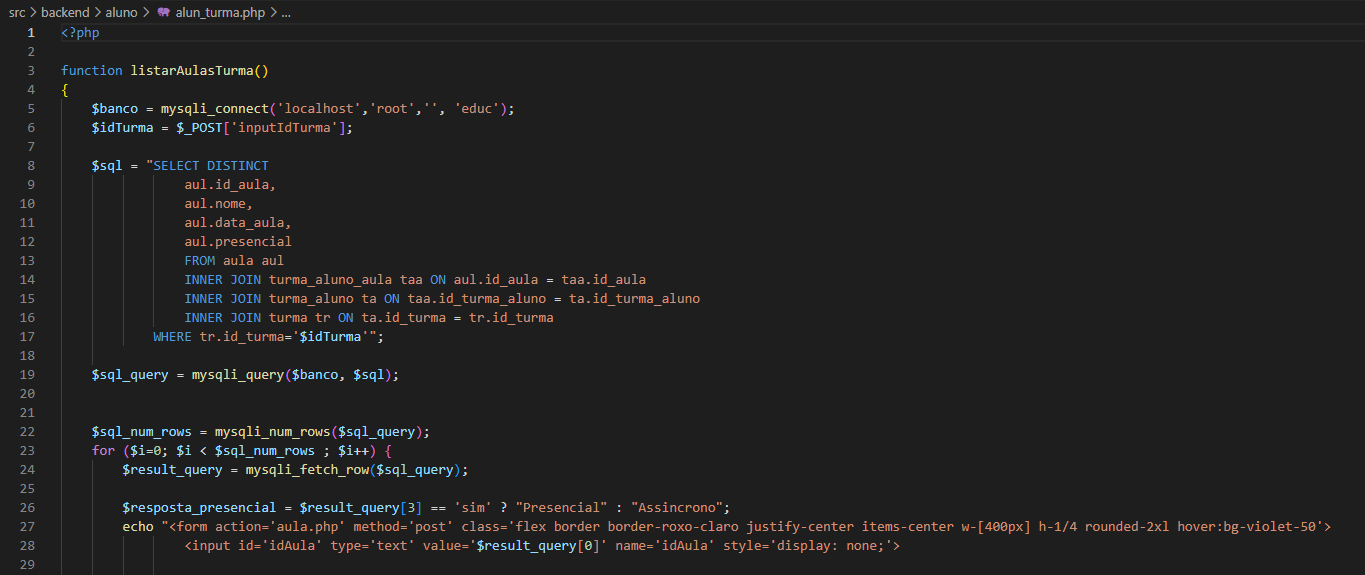
Fonte: Autor

Imagem 31 - Código da Parte Turma.

Fonte: Autor

### 10.3.1 Descrição de algumas funções no código do Aluno

#### alun\_enviar\_… - Código responsável por enviar atividades realizadas pelo aluno.

#### alun\_listar\_ativ… - Código responsável por listar as atividades disponíveis para o aluno.

#### alun\_listar\_cont… - Código responsável por listar os conteúdos assíncronos disponíveis para o aluno, permitindo o download dos arquivos dos conteúdos.

#### alun\_baixar\_… - Código responsável por realizar o download de um arquivo.

#### alun\_dados\_… - Código responsável por listar os dados do perfil do aluno.

#### alun\_listar\_turm… - Código responsável por listar as turmas nas quais o aluno está registrado.

#### alun\_listar\_aul… - Código responsável por listar as aulas de uma turma específica.

## 10.4 Diagrama de Classe

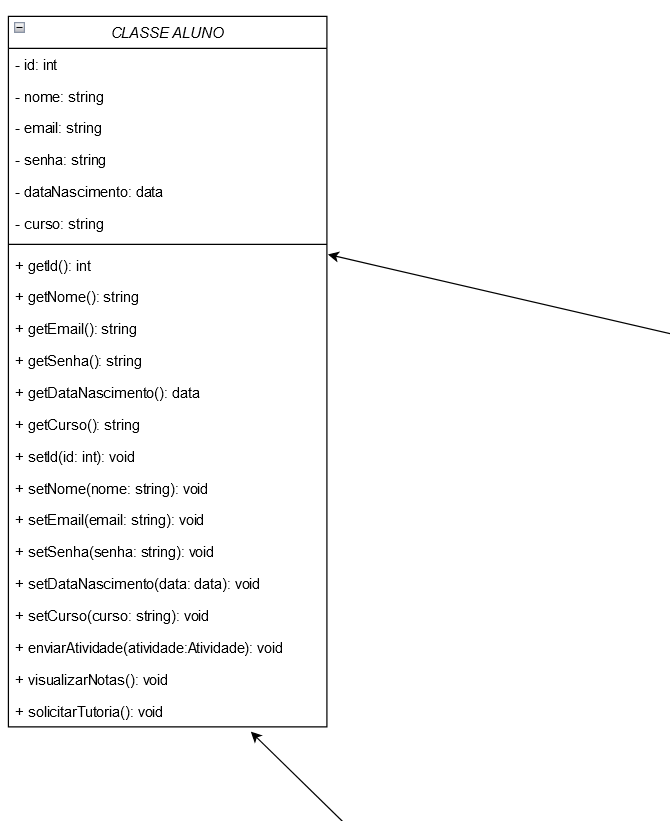


Imagem 32 – Imagem com Zoom da Classe Aluno.

Fonte: Autor

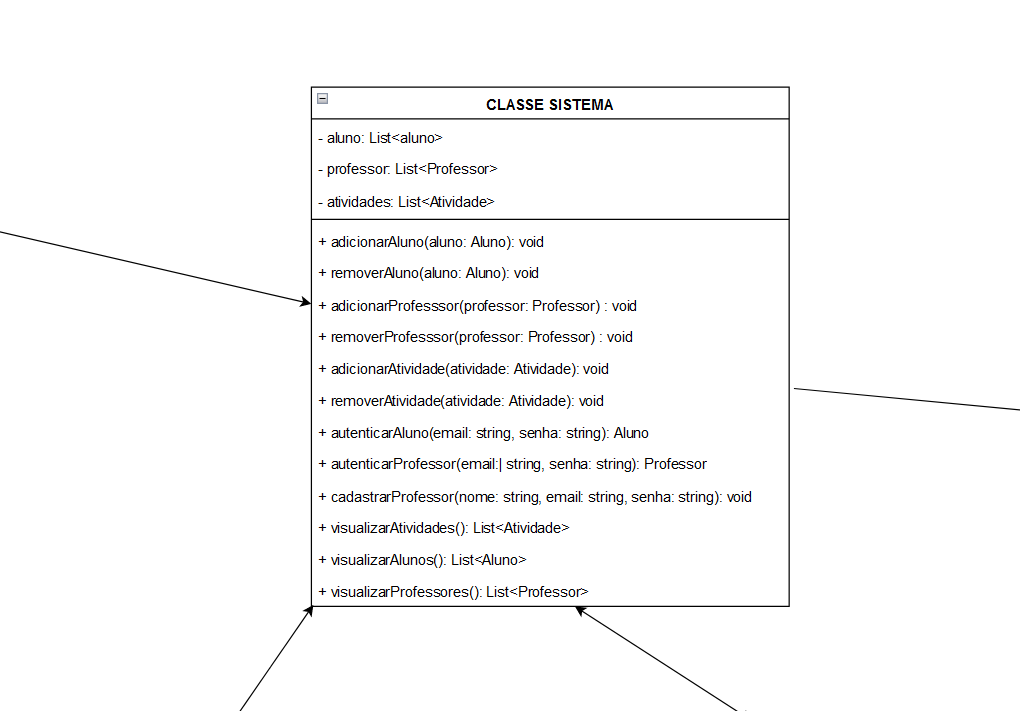


Imagem 33 – Imagem com Zoom da classe Sistema.

Fonte: Autor

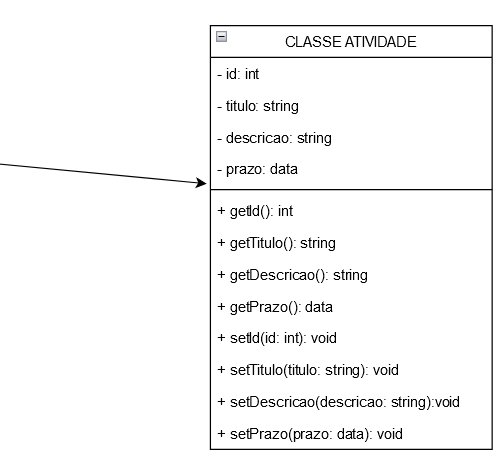


Imagem 34 – Imagem com Zoom da Classe Atividade.

Fonte: Autor

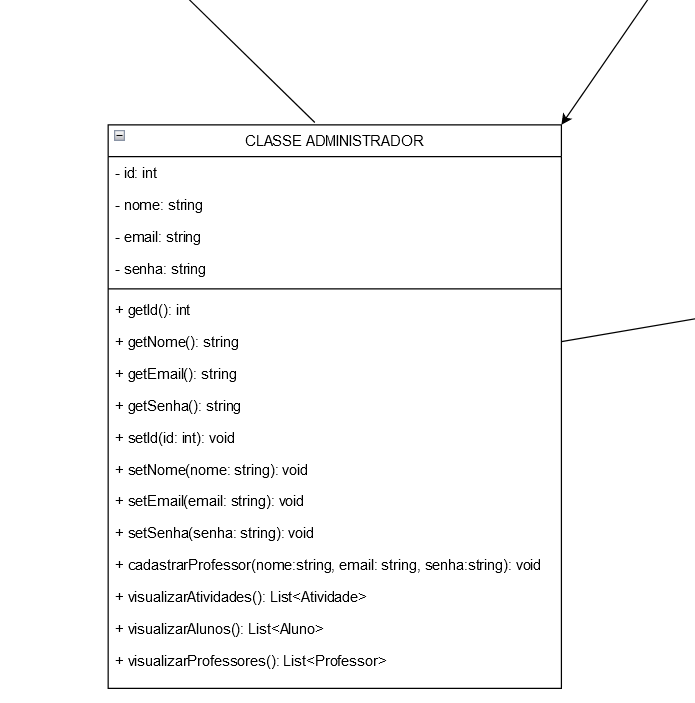


Imagem 35 – Imagem com Zoom da classe Administrador.

Fonte: Autor

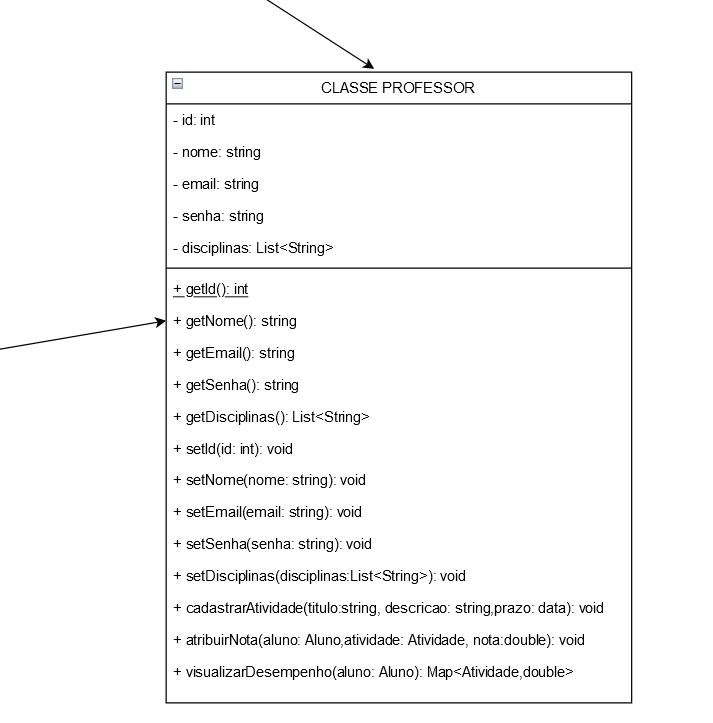


Imagem 36 – Imagem com Zoom da classe Professor.

Fonte: Autor