

UNIVERSIDADE FEDERAL DO PARANÁ

GUILHERME ARTHUR NUNES MENEGARI

JULIA VALERANOVICZ DE OLIVEIRA

PEDRO AUGUSTO LEMOS

SOPHIA COSTA SOARES E SILVA

PROJETO WEB

SISTEMA ACADEMICO DE GERENCIAMENTO UNIVERSITÁRIO
(SAGU)

CURITIBA

2024

GUILHERME ARTHUR NUNES MENEGARI

JULIA VALERANOVICZ DE OLIVEIRA

PEDRO AUGUSTO LEMOS

SOPHIA COSTA SOARES E SILVA

PROJETO WEB

SISTEMA ACADEMICO DE GERENCIAMENTO UNIVERSITÁRIO
(SAGU)

Trabalho apresentado a disciplina
DS122 – Desenvolvimento Web I
Desenvolvimento de um sistema
web de postagem de documentos
acadêmicos e gerenciamento de
alunos.

Prof. Dr. Roberson César Araujo

CURITIBA

2024

PREFACIO	5
1. Introdução	6
1.1 Tema	6
1.2 Objetivo	6
1.3 Delimitação do Problema	6
1.4 Escolha do Tema	6
1.5 Método de Trabalho	7
1.6 Organização do Trabalho.....	7
2. Descrição geral do sistema	8
2.1 Descrição do problema.....	8
2.2 Principais Envolvidos e Suas Características	8
2.2.1 Usuários do Sistema	8
2.1.2 Desenvolvedores do Sistema.....	8
2.3 Regras de Negócios	8
3. Requisitos do Sistema	10
3.1 Prioridade dos Requisitos.....	10
3.1.1 Prioridades dos Requisitos Funcionais	10
3.1.2 Prioridades dos Requisitos Não-Funcionais.....	11
3.2 Requisitos Funcionais.....	11
3.3 Requisitos não funcionais	13
3.3 Diagrama de Caso de Uso	14
3.4 Prototipação de telas.....	15
4. Análise e Design	18
4.1 Arquitetura do Sistema	18
4.2 Modelo do Domínio.....	18
4.3 Modelo de Dados.....	19
4.3.1 Modelo Lógico do Banco de Dados.....	19
4.3.2 Modelo Físico do Banco de Dados.....	19
4.3.3 Dicionário do Banco de Dados	21
Tabela: student.....	21
Tabela: teacher.....	22
Tabela: user.....	22

Tabela: documents	22
Tabela: folder	22
4.4 Ambiente de Desenvolvimento	23
5. Implementação	24
6. Manual de Implantação	26
6.1 Pré-Requisitos.....	26
6.2 Baixe e instale o Node.js:	26
6.3 Instalação do MySQL	26
6.4 Configuração do Projeto.....	27
6.5 Executando Migrações	28
6.6 Gerando Migrações.....	28
7. Testes.....	29
8. Manual do Usuário.....	32
9. Conclusão e considerações finais	34

PREFACIO

Este documento tem como objetivo apresentar e documentar o Sistema Acadêmico de Gerenciamento Universitário (SAGU), para facilitar o entendimento e a manutenção deste sistema, que tem como base um programa web. Aqui será apresentado todos os requisitos funcionais e não funcionais do projeto, além de suas representações em UML e anotações de testes unitários.

Uma navegação mais ágil pode ser executada a partir do uso do sumário clicando na área desejada, alguns itens estão descritos no glossário ao fim do documento para fácil entendimento.

Este projeto está sendo realizado sobre solicitação e supervisão do professor Roberson César Araujo, na disciplina de Desenvolvimento Web I.

Modelo de documentação baseado em “Documentação de um produto de Software” por Prof.^a Ana Paula Gonçalves Serra.

1. Introdução

1.1 Tema

Este documento tem como objetivo apresentar o Sistema Acadêmico de Gerenciamento Universitário (SAGU), um sistema Web que permite postagem de pastas (Públicas ou Privadas), de Documentos Acadêmicos, por Aluno que podem se cadastrar no mesmo, há também professores que agem como Administradores do sistema podendo administrar os alunos cadastrados.

1.2 Objetivo

O Objetivo do projeto é dar um ambiente para postagem de conteúdos acadêmicos que possibilite trocas de apostilas ou outros arquivos para ajudar em estudos ou em projetos colaborativos entre os alunos, tendo todo esse processo sendo supervisionado por Professores cadastrados no sistema.

1.3 Delimitação do Problema

O Escopo do problema se expande a partir do ponto em que se vê a dificuldade de um ponto para compartilhar projetos e informações confiáveis sobre campos do conhecimento, com uma administração competente e ativa que pode monitorar todo o processo de estudos e conduta dos usuários, de uma forma simplificada e intuitiva, que permita que essas ações sejam realizadas com poucos cliques.

1.4 Escolha do Tema

Este foi um dos temas apresentados pelo professor da disciplina, tema esse que chamou a atenção dos envolvidos no projeto pela flexibilidade que teria para implementar os conteúdos apresentados na disciplina de diversas formas, e criar uma boa aplicação de comunicação cliente servidor em um projeto full-stack.

1.5 Método de Trabalho

O projeto foi organizado através de diagramas de uml e construções lógicas do banco de dados e foi implementado através de orientado a objetos utilizando as tecnologias de HTML, Javascript, CSS(Tailwind), TypeScript, PHP e MySQL. Usando os frameworks React para front-end e Node em parte do back-end.

1.6 Organização do Trabalho

O Documento está organizado em 9 partes principais, que estão todas subdivididas em alguns tópicos. As partes principais são, Introdução, Descrição Geral do Sistema, Requisitos do Sistema, Análise e Design, Implementação, Testes, Implantação, Manual do Usuário e Conclusões e Considerações finais. Tendo disponível para facilitar a compreensão um sumario e um glossário.

2. Descrição geral do sistema

2.1 Descrição do problema

O problema analisado para este sistema seria da falta de um ambiente com boa administração em que alunos possam compartilhar dados acadêmicos que tenham informações pertinentes. Esse problema afeta principalmente os estudantes que não têm a possibilidade de ter um ambiente para essas interações, sendo assim o SAGU foi desenvolvido como um ambiente estável e intuitivo com constante supervisão de professores que podem administrar os conteúdos do sistema no geral, criando um ambiente controlado para essas atividades.

2.2 Principais Envolvidos e Suas Características

2.2.1 Usuários do Sistema

Os Usuários do SAGU se resumem aos alunos cadastrados, sendo esse um sistema livre que pode ser utilizado por qualquer estudante em busca de conteúdo para seus estudos de forma gratuita, em uma comunidade que compartilha estes materiais para usos públicos, além de exibirem seus projetos pessoais.

2.1.2 Desenvolvedores do Sistema

Este sistema foi desenvolvido por 4 Alunos da Universidade Federal do Paraná Guilherme Arthur Nunes Menegari, Julia Valeranovicz de Oliveira, Pedro Augusto Lemos e Sophia Costa Soares e Silva, sob solicitação do Prof. Roberson César Araujo.

2.3 Regras de Negócios

R1 – Exclusividade de E-mail:
Descrição: Garantir que cada endereço de e-mail cadastrado no sistema seja único, evitando a duplicação de contas de usuário.

R2 – Criptografar Senhas:

Descrição: As senhas dos usuários devem ser armazenadas no banco de dados de forma criptografada, garantindo a segurança dos dados em caso de acesso não autorizado.
--

R3 – Professores/Admins:

Descrição: Certificar-se de que apenas professores tenham acesso a determinadas funcionalidades ou áreas restritas do sistema, garantindo a segurança e a integridade das informações sensíveis.
--

3. Requisitos do Sistema

3.1 Prioridade dos Requisitos

3.1.1 Prioridades dos Requisitos Funcionais

RF1: Cadastro de alunos e professores - Alta
RF2: Sistema de login/logout - Alta
RF3: Recuperação de senha e gerenciamento de perfil de usuário - Alta
RF4: Suporte para upload de diferentes tipos de arquivos - Alta
RF5: Organização de arquivos em pastas específicas do projeto - Média
RF6: Possibilidade de professores monitorarem pastas - Média
RF7: Sistema de gerenciamento de alunos por professores - Baixa
RF8: Funcionalidade para alunos criarem e personalizarem seus portfólios digitais - Baixa
RF9: Opções para incluir projetos, certificados, vídeos, imagens e documentos - Baixa

3.1.2 Prioridades dos Requisitos Não-Funcionais

RNF1: Segurança - Alta
RNF2: Desempenho - Alta
RNF3: Escalabilidade - Alta
RNF4: Compatibilidade - Média
RNF5: Usabilidade - Alta
RNF6: Manutenibilidade - Média
RNF7: Confiabilidade - Alta

3.2 Requisitos Funcionais

RF1: Cadastro de alunos e professores
- Descrição: Este requisito envolve a capacidade de registrar alunos e professores no sistema, fornecendo informações relevantes para cada tipo de usuário.

RF2: Sistema de login/logout
- Descrição: Um sistema de autenticação que permite aos usuários acessarem o sistema e saírem dele quando desejarem.

RF3: Recuperação de senha e gerenciamento de perfil de usuário
- Descrição: Permite aos usuários recuperarem suas senhas de forma segura e gerenciarem suas informações de perfil.

RF4: Suporte para upload de diferentes tipos de arquivos (PDF, DOCX, PPT, MP4, etc.)
- Descrição: Capacidade de fazer upload de diversos tipos de arquivos para o sistema, como documentos, apresentações e vídeos.

RF5: Organização de arquivos em pastas específicas do projeto

- Descrição: Permite aos usuários organizar os arquivos em pastas relacionadas a projetos específicos, facilitando a gestão e o acesso.

RF6: Possibilidade de professores monitorarem pastas
--

- Descrição: Permite que os professores monitorem pastas específicas para acompanhar o progresso e o trabalho dos alunos.

RF7: Sistema de gerenciamento de alunos por professores

- Descrição: Ferramentas que permitem aos professores gerenciarem os alunos e seus perfis

RF8: Funcionalidade para alunos criarem e personalizarem seus portfólios digitais

- Descrição: Permite aos alunos criar e personalizar portfólios digitais para exibir seu trabalho e realizações.
--

RF9: Opções para incluir projetos, certificados, vídeos, imagens e documentos

- Descrição: Oferece aos alunos opções diversificadas para incluir vários tipos de conteúdo em seus portfólios digitais.
--

3.3 Requisitos não funcionais

RNF1: Segurança

- Descrição: O sistema deve garantir a segurança dos dados dos usuários incluindo medidas como criptografia de dados.

RNF2: Desempenho

- Descrição: O sistema deve ser capaz de lidar com múltiplos usuários simultaneamente sem comprometer o desempenho, garantindo tempos de resposta rápidos mesmo durante picos de uso.

RNF3: Escalabilidade

- Descrição: Deve ser possível escalar o sistema de forma eficiente para lidar com um aumento no número de usuários ou no volume de dados, sem que isso cause degradação no desempenho.

RNF4: Compatibilidade

- Descrição: O sistema deve ser compatível com uma variedade de telas e navegadores, garantindo uma experiência consistente para os usuários independentemente do dispositivo que estiverem usando.

RNF5: Usabilidade

- Descrição: O sistema deve ser intuitivo e de fácil utilização, com uma interface amigável que permita aos usuários navegar e utilizar as funcionalidades com facilidade, mesmo sem treinamento prévio.

RNF6: Manutenibilidade

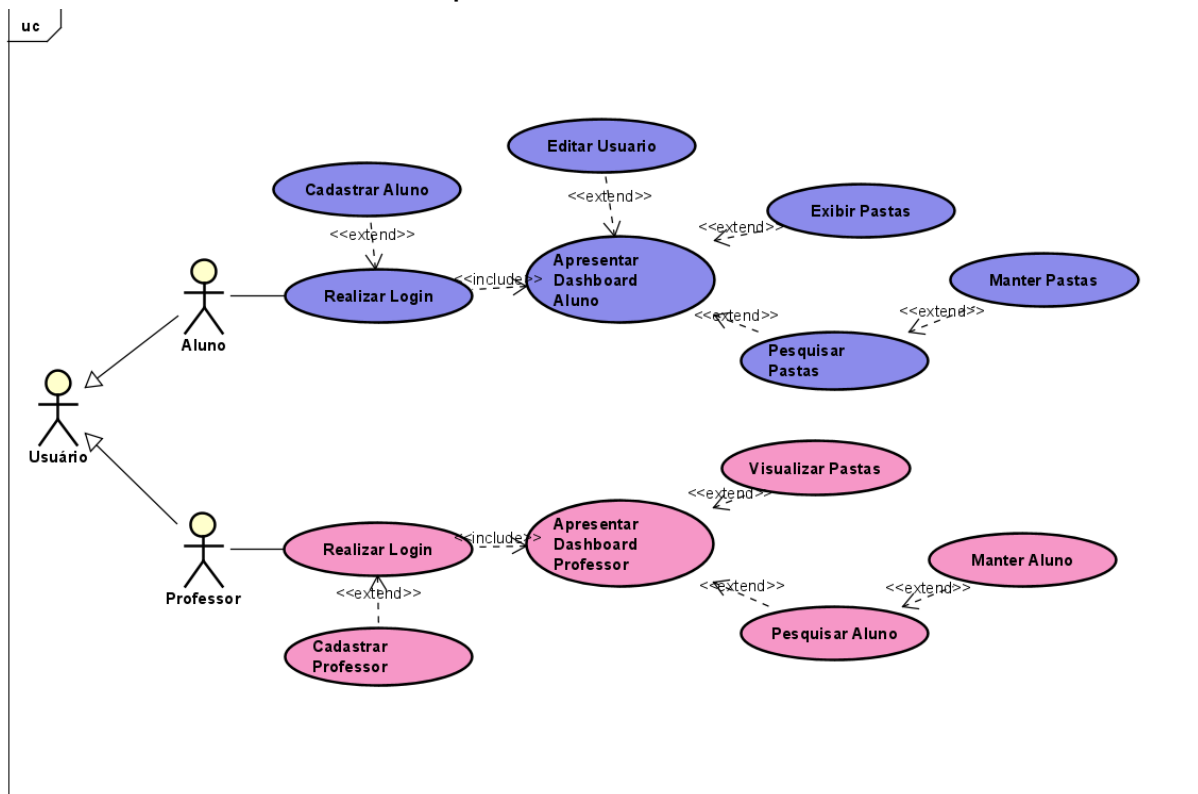
- Descrição: O código do sistema deve ser bem documentado e modularizado, facilitando futuras atualizações, correções de bugs e incorporação de novas funcionalidades.

RNF7: Confiabilidade

- Descrição: O sistema deve ser robusto e confiável, minimizando a ocorrência de erros e falhas que possam afetar a experiência do usuário.

3.3 Diagrama de Caso de Uso

Aqui é apresentado o diagrama que apresenta as interações dos usuários com o sistema dividido em partes.

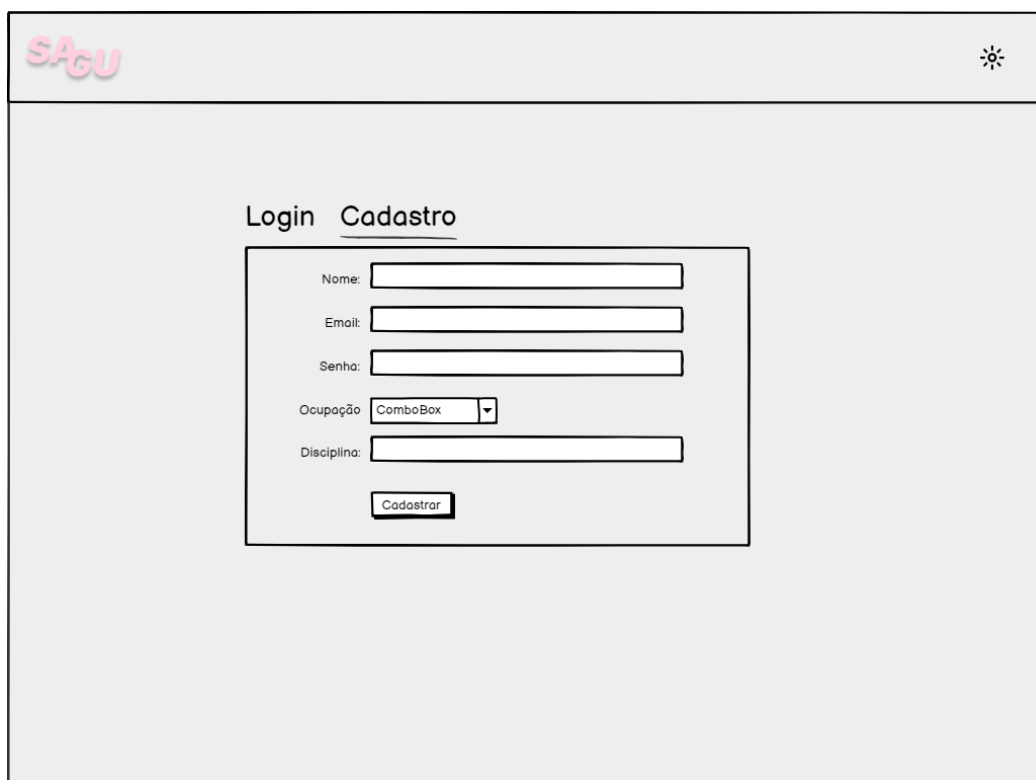


3.4 Prototipação de telas




The image shows a wireframe of a login screen. At the top left is a pink logo with the text 'SAGU'. At the top right is a sun-like icon. Below the header, there are two tabs: 'Login' (which is underlined) and 'Cadastro'. A rectangular box contains the login form, which includes an 'Email:' label followed by a text input field, a 'Senha:' label followed by a text input field, and a 'Login' button at the bottom.


Tela de Login – Seguindo os RF2, RNF1 e RNF5



The image shows a wireframe of a registration screen. At the top left is a pink logo with the text 'SAGU'. At the top right is a sun-like icon. Below the header, there are two tabs: 'Login' and 'Cadastro' (which is underlined). A rectangular box contains the registration form, which includes a 'Nome:' label followed by a text input field, an 'Email:' label followed by a text input field, a 'Senha:' label followed by a text input field, an 'Ocupação' label followed by a dropdown menu showing 'ComboBox', and a 'Disciplina:' label followed by a text input field. A 'Cadastrar' button is located at the bottom of the form.

Tela de Cadastro - Seguindo os RF1, RNF1 e RNF5

 Bem vindo, aluno

 [Editar Usuário](#) [Sair](#)

Pastas

Exemplo de pasta

Documento.pdf

Excluir

Documento.pdf

Excluir

Inserir documento

Escolher ficheiro

Nenhum ficheiro selecionado

Adicionar


Exemplo de pasta


Visualizar documentos

Exemplo de pasta

Visualizar documentos

Dashboard Aluno – Seguindo RF3, RF4, RF5, RF8, RF9

 Bem vindo, professor

 [Editar Usuário](#) [Sair](#)

Lista de Usuários

nome_usuario
example@email.com

Editar Excluir

nome_usuario2
example@email.com

Editar Excluir

nome_usuario3
example@email.com

Editar Excluir

nome_usuario4
example@email.com

Editar Excluir

nome_usuario5
example@email.com

Editar Excluir

Lista de Documentos

documento.pdf
nome_usuario

Download

arquivo.c
nome_usuario4

Download

imagem.img
nome_usuario5

Download

Dashboard Professor – Seguindo os RF3, RF6 e RF7

SAGU Bem vindo, professor

Meu Perfil

Sair

Lista de Usuários

nome_usuario
example@email.com

Editar

Excluir

nome_usuario2
example@email.com

Editar

Excluir

nome_usuario3
example@email.com

Editar

Excluir

nome_usuario4
example@email.com

Editar

Excluir

nome_usuario5
example@email.com

Editar

Excluir

Lista de Documentos

documento.pdf
nome_usuario

arquivo.c
nome_usuario4

imagem.png
nome_usuario5

Editar informações

Nome

Placeholder

Email

Placeholder@email.com

Senha

Salvar alterações

Modal de Edição de Logado – Seguindo RF3

SAGU Bem vindo, professor

Editar Usuário

Sair

Editar usuário

Nome:

Email:

Senha:

Salvar

Voltar

Gerenciamento de Usuário – Seguindo RF1, RF3 e RF7

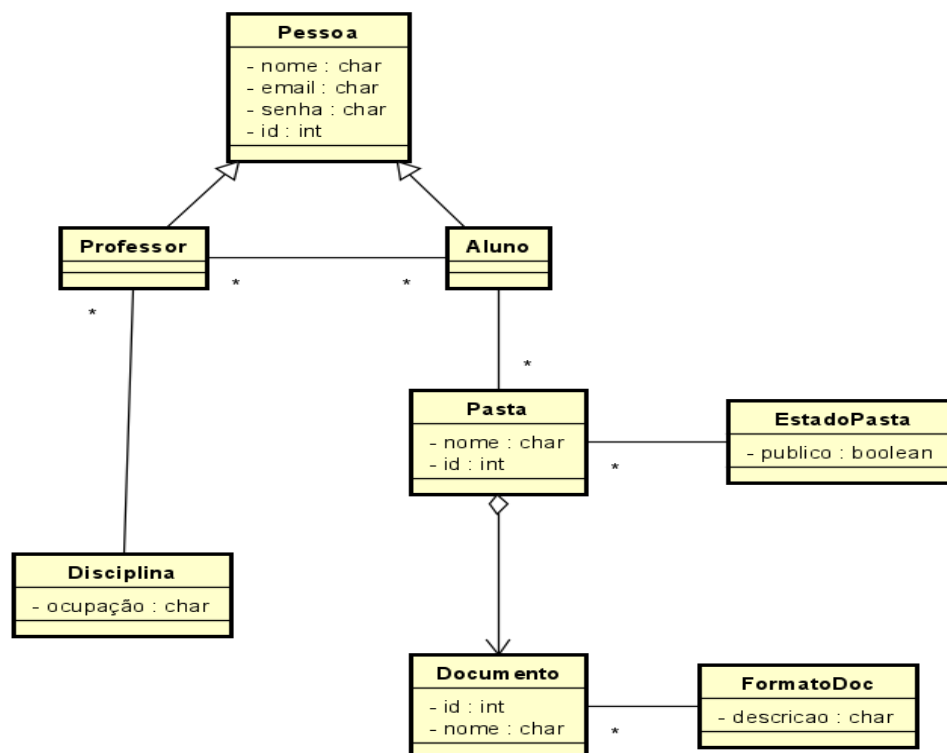
4. Análise e Design

4.1 Arquitetura do Sistema

O sistema é projetado com uma arquitetura cliente-servidor, que utiliza como Banco de Dados o MySQL2, composta por três camadas principais: Interface do Usuário, Lógica de Aplicação e Acesso a Dados. Além disso, são estabelecidos requisitos não funcionais para garantir a segurança, desempenho, escalabilidade, compatibilidade, usabilidade, manutenibilidade e confiabilidade do sistema. Utilizando uma configuração de hardware com um servidor que roda na mesma máquina que o próprio sistema, o sistema de armazenamento também é feito na máquina através de um ssd. Já a rede é configurada em LAN, devido ao escopo e objetivos do projeto.

4.2 Modelo do Domínio

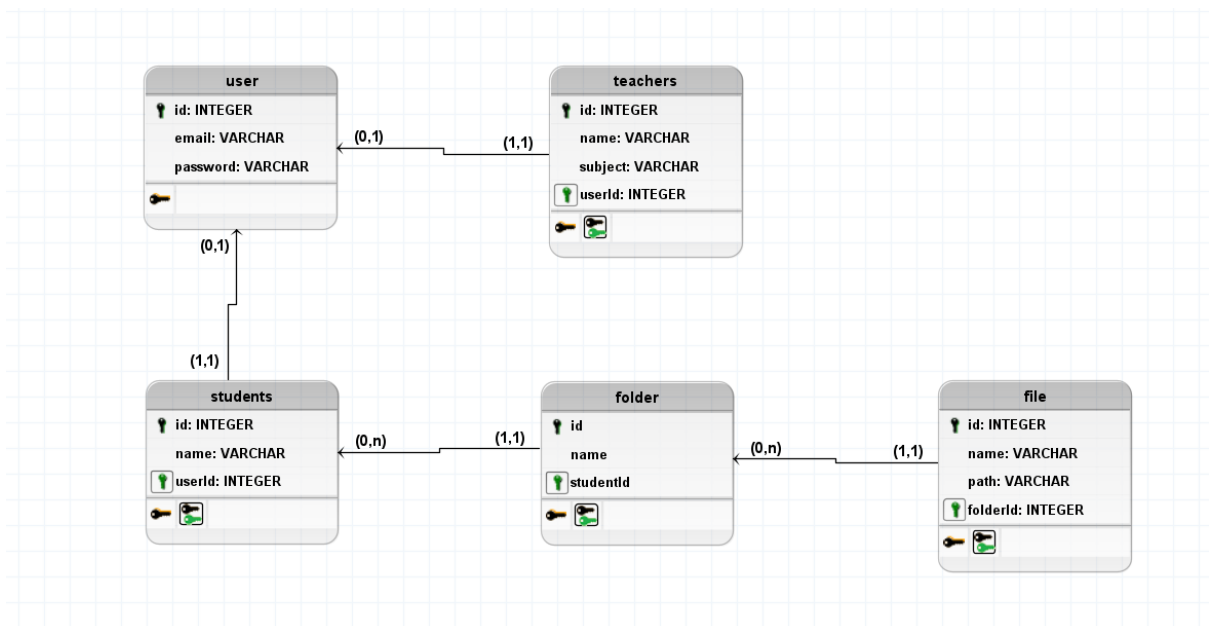
Esse é o Diagrama de classes que dita a arquitetura inicial do projeto, ele representa todos os objetos atuantes no sistema.



4.3 Modelo de Dados

4.3.1 Modelo Lógico do Banco de Dados

Abaixo é apresentado o modelo lógico do banco de dados utilizado no processo feito através da ferramenta brModelo.



4.3.2 Modelo Físico do Banco de Dados

Abaixo se encontra o modelo físico do banco de dados (Seus scripts) na linguagem SQL no ambiente MySQL workbench.

Create database projetoSagu;

```
CREATE TABLE `student` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `userId` int NOT NULL,  
  PRIMARY KEY (`id`),
```

```
FOREIGN KEY (`userId`) REFERENCES `user`(`id`) ON  
DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB;
```

```
CREATE TABLE `teacher` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `subject` varchar(255) NOT NULL,  
  `userId` int NOT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`userId`) REFERENCES `user`(`id`) ON  
DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB;
```

```
CREATE TABLE `user` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `email` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  UNIQUE INDEX `IDX_e12875dfb3b1d92d7d7c5377e2`  
(`email`),  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

```
CREATE TABLE `documents` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `path` varchar(255) NOT NULL,  
  `folderId` int NOT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`folderId`) REFERENCES `folder`(`id`) ON  
DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB;
```

```
CREATE TABLE `folder` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,
```

```
`studentId` int NOT NULL,  
PRIMARY KEY (`id`),  
FOREIGN KEY (`studentId`) REFERENCES `student` (`id`)  
ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB;
```

```
ALTER TABLE `folder` DROP FOREIGN KEY  
`FK_a51d04cc7cd9e5ca00a28d75e85`;  
ALTER TABLE `documents` DROP FOREIGN KEY  
`FK_cf0a9fa48053d1f93da40713cc1`;  
ALTER TABLE `teacher` DROP FOREIGN KEY  
`FK_4f596730e16ee49d9b081b5d8e5`;  
ALTER TABLE `student` DROP FOREIGN KEY  
`FK_b35463776b4a11a3df3c30d920a`;
```

```
DROP TABLE `folder`;  
DROP TABLE `documents`;  
DROP INDEX `IDX_e12875dfb3b1d92d7d7c5377e2` ON  
`user`;  
DROP TABLE `user`;  
DROP TABLE `teacher`;  
DROP TABLE `student`;
```

4.3.3 Dicionário do Banco de Dados

Tabela: student

id: Um identificador único para cada registro na tabela de alunos.

name: O nome do aluno.

userId: Uma chave estrangeira que faz referência ao ID do usuário na tabela de usuários. Este campo está relacionado ao usuário que possui a conta do aluno.

Tabela: teacher

id: Um identificador único para cada registro na tabela de professores.

name: O nome do professor.

subject: O assunto ou disciplina que o professor ensina.

userId: Uma chave estrangeira que faz referência ao ID do usuário na tabela de usuários. Este campo está relacionado ao usuário que possui a conta do professor.

Tabela: user

id: Um identificador único para cada registro na tabela de usuários.

email: O endereço de e-mail único associado ao usuário.

password: A senha criptografada do usuário.

Tabela: documents

id: Um identificador único para cada registro na tabela de documentos.

name: O nome do documento.

path: O caminho ou localização do documento.

folderId: Uma chave estrangeira que faz referência ao ID da pasta na tabela de pastas. Este campo está relacionado à pasta na qual o documento está armazenado.

Tabela: folder

id: Um identificador único para cada registro na tabela de pastas.

name: O nome da pasta.

studentId: Uma chave estrangeira que faz referência ao ID do aluno na tabela de alunos. Este campo está relacionado ao aluno que possui a pasta.

4.4 Ambiente de Desenvolvimento

O Projeto é desenvolvido nas linguagens, HTML, JavaScript, TypeScript, PHP e CSS(Tailwind). Utiliza-se também dois frameworks de js, O React e Node.js, em questão de banco de dados ele é escrito em SQL através do MySQL Workbench 8.0. O Ambiente gráfico utilizado é o Visual Studio Code que é disponibilizado gratuitamente pela Microsoft. Hardwares e redes variam, pois, são utilizados os locais dos desenvolvedores.

5. Implementação

O sistema foi desenvolvido utilizando o framework React, conhecido por sua capacidade de dividir as funcionalidades do projeto em componentes, o que facilita a organização, leitura e manutenção do código ao longo do tempo. Além disso, a utilização do subtipo TypeScript proporciona códigos mais específicos e legíveis, aumentando a segurança e a produtividade durante o desenvolvimento. Para a parte de estilização foi utilizado o Tailwind uma biblioteca de componentes de CSS sendo muito útil para a implementação de um visual limpo e responsivo.

No diretório do front-end, as funcionalidades foram organizadas em duas principais categorias:

Páginas e Serviços: As páginas representam as diferentes interfaces de usuário da aplicação, enquanto os serviços são responsáveis pela integração com o backend e pela lógica de negócios da aplicação.

App: Este diretório é utilizado para configurar as rotas da aplicação, conectando-as às funcionalidades do backend.

No lado do backend, a estrutura do projeto foi organizada em três principais componentes:

Controladores: Responsáveis por definir as ações que serão executadas em resposta às requisições HTTP recebidas pelo servidor.

Repositórios: Responsáveis por interagir com o banco de dados, realizando operações de leitura e escrita nos dados armazenados.

Rotas: Uma página dedicada à definição das rotas da API RESTful, mapeando as requisições HTTP para os controladores correspondentes.

A segurança do sistema é garantida através do hashing das informações do usuário, o que protege as credenciais armazenadas no banco de dados. Além disso, o sistema utiliza o JWT (JSON Web Tokens) para gerenciar as sessões de usuário, fornecendo autenticação segura e autorização de acesso às funcionalidades do sistema.

Na questão de servidor ele utiliza o TypeORM para gerar migrações novas do banco de dados em SQL, necessitando apenas que exista uma database chamada projetoSagu.

6. Manual de Implantação

Para a implantação do sistema de forma correta em um ambiente local siga os passos a seguir:

6.1 Pré-Requisitos

Antes de começar, certifique-se de ter instalado os seguintes componentes em seu sistema:

- [Node.js](https://nodejs.org/)
- [npm](https://www.npmjs.com/) (normalmente instalado automaticamente com o Node.js)
- [MySQL](https://www.mysql.com/)

6.2 Baixe e instale o Node.js:

- Acesse [Node.js Downloads](https://nodejs.org/en/download/) e escolha a versão adequada para seu sistema operacional.
- Siga as instruções de instalação fornecidas pelo instalador.

6.3 Instalação do MySQL

6.3.1 Baixe e instale o MySQL:

- Acesse [MySQL Downloads](https://dev.mysql.com/downloads/) e escolha a versão adequada para seu sistema operacional.
- Siga as instruções de instalação, definindo uma senha para o usuário root quando solicitado.

6.3.2 Configurar o MySQL Workbench (opcional):

- O MySQL Workbench pode ser utilizado para gerenciar visualmente o banco de dados. É útil para criar esquemas, executar consultas e outras tarefas administrativas.

- Baixe o MySQL Workbench
[aqui](<https://dev.mysql.com/downloads/workbench/>).

6.4 Configuração do Projeto

6.4.1 Clonar o repositório:

```
git clone https://github.com/aishspo/ProjetoFinal-DS122
```

6.4.2 Instalar dependências:

```
cd frontend
```

```
npm install
```

```
cd ..
```

```
cd backend
```

```
npm install
```

6.4.3 Criar base de dados

- Abra o MySQL Workbench, acesse a conexão criada durante a configuração do mesmo, crie uma base de dados e a nomeie conforme o desejado (nesse caso, utilizamos “projetoSagu”).

6.4.4 Configurar variáveis de ambiente:

- Renomeie o arquivo ``.env.example`` para ``.env``.

- Abra o arquivo ``.env`` e configure as variáveis de ambiente de acordo com o seu ambiente de desenvolvimento, especificando as credenciais do banco de dados.

6.4.5 Configurar Xampp:

Inserir arquivo erro500.php na pasta `htdocs` do Xampp e iniciá-lo em modo Apache.

6.5 Executando Migrações

Antes de iniciar a aplicação, é necessário configurar o banco de dados executando as migrações:

6.5.1 Executar Migrações:

```
npm migration:run
```

```
npm run dev
```

Isso irá iniciar o servidor usando `nodemon` e `ts-node` para hot reloading.

6.6 Gerando Migrações

Quando realizar mudanças nas entidades ou alterar o esquema do banco de dados, você pode gerar uma nova migração automaticamente com o comando:

```
npm migration:generate
```

Este comando irá gerar um arquivo de migração na pasta `src/migrations` com as alterações necessárias para aplicar ao banco de dados.

7. Testes

Neste capítulo será apresentado os testes realizados para avaliar a viabilidade e confiabilidade do sistema, utilizando aqui diversas ferramentas e cenários no modelo

Nº do teste

Finalidade

Resultado

Nº do teste. 1

Finalidade. Teste de Cadastro (Aluno)

Resultado. O Aluno é cadastrado com sucesso e caso faltem campos apresenta uma mensagem de erro.

Nº do teste. 2

Finalidade. Teste de Cadastro (Professor)

Resultado. O Professor é cadastrado com sucesso e caso haja campos necessários para apresentar uma mensagem de erro.

Nº do teste. 3

Finalidade. Login

Resultado. Se realizado o login por alguém identificado como aluno ele o envia para sua dashboard correta, já se for professor para uma dashboard diferente.

Nº do teste. 4

Finalidade. Testar a inclusão de formatos diferentes de arquivos.

Resultado. Com o teste foi possível incluir diversos tipos de arquivos. (.jpg, .gif, .png, .pdf, .docx, .mp4)

Nº do teste. 5

Finalidade. Excluir Arquivos

Resultado. É possível excluir qualquer arquivo inserido.

Nº do teste. 6

Finalidade. Criar e Excluir Pastas

Resultado. É possível criar e excluir pastas a qualquer momento. (Incluindo seus documentos)

Nº do teste. 7

Finalidade. Editar Perfil

Resultado. Para professor e aluno foi possível editar todos os campos dos próprios perfis, sem ser removido do site.

Nº do teste. 8

Finalidade. Gerenciamento de Alunos (Professor)

Resultado. É possível como professor remover qualquer aluno ou editar suas características, sendo que ao ser removido o aluno tem todas as suas pastas removidas com ele.

Nº do teste. 9

Finalidade. Download de arquivos

Resultado. Foi possível que qualquer tipo de usuário baixe documentos postados.

Nº do teste. 10

Finalidade. E-mails Únicos

Resultado. É impossível criar qualquer usuário com um Email de usuário já ativo.

8. Manual do Usuário

Neste Capítulo está contido as informações pertinentes para uso do sistema para ambos os tipos de usuários.

É importante que para acessar o sistema e fazer com que ele seja executado corretamente é necessário seguir os passos listados no capítulo 6. Manual de Implementação. Esse Manual é feito com base em que o sistema já esteja rodando.

Início do Sistema:

Com o sistema rodando acesse a página <http://localhost:5173>

Isso o enviará diretamente a página de login.

Para o primeiro acesso é necessário um cadastro então acesse “Cadastrar”

Nesta aba devem ser preenchidos todos os campos, se incluído que sua ocupação é professor você deverá preencher a opção disciplina.

Clique no botão de finalizar cadastro.

Login:

Estando na tela de login o usuário deve inserir seu e-mail e sua senha e clicar em “entrar”.

Caso a senha tenha sido esquecida é necessário avisar a um administrador para que alterem sua senha. (Isso pode ser feito através de contato via e-mail)

Dashboard (Aluno)

Ao logar como aluno você tem opções de criar, editar, visualizar e excluir pastas.

As pastas exibidas podem ser vistas entre as suas pastas e as pastas de outros alunos que se encontram para o público visualizar.

Ao clicar no ícone da pasta você vai a acessar tendo a opção de (se a pasta lhe pertence) remover e adicionar documentos nas mesmas.

Outra opção a se notar é de editar usuário o que permite a mudança de nome, e-mail e senha do usuário logado.

Dashboard (Professor)

O Professor tem acesso a uma tabela para editar ou remover onde pode gerenciar os alunos cadastrados, funcionando como administrador, ao clicar para remover um usuário é dada uma confirmação e ele é removido do Banco de Dados.

Ao editar, apresenta um formulário com os dados do aluno em que se pode editar: nome, e-mail e senha, com eles logados ou não.

É possível também o acesso a todas as pastas cadastradas no sistema para realizar um monitoramento.

Na Página de professor há a opção de editar usuário onde é possível alterar os dados do próprio professor logado.

Logout

Ambos têm em suas dashboards a opção logout onde se clicada finaliza a sessão e o token utilizado para ela.

9. Conclusão e considerações finais

Os resultados obtidos destacam a viabilidade e utilidade do sistema para os usuários finais, incluindo alunos, professores/Administradores. No entanto, durante o processo de desenvolvimento, algumas limitações foram identificadas, como a questão de armazenamento e o fato da limitação do próprio hardware em que o sistema está sendo rodado atualmente. Necessitando de um hardware mais potente e maior armazenamento para facilitar a escalabilidade do projeto.

A inovação do sistema reside na sua capacidade de proporcionar uma plataforma colaborativa e organizada para o compartilhamento e gerenciamento eficiente de documentos no contexto educacional. Além disso, o sistema possui potencial para integração com outras plataformas educacionais ou de gerenciamento de conteúdo, ampliando suas funcionalidades e oferecendo uma experiência mais abrangente aos usuários.

Como trabalho futuro, sugere-se continuar o desenvolvimento do sistema para implementar novas funcionalidades, como ferramentas de avaliação e integrações com sistemas de videoconferência, além de realizar testes extensivos para garantir a estabilidade e confiabilidade do sistema em ambientes de produção. Essas considerações finais destacam a importância do sistema desenvolvido e seu potencial de contribuição para o ambiente educacional e de colaboração.