

**CENTRO PAULA SOUZA**  
**FACULDADE DE TECNOLOGIA DE FRANCA**  
**“Dr. THOMAZ NOVELINO”**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE**  
**SISTEMAS**

**GUILHERME HAKIME ANDRADE**

**EDUSTREAM: UMA PLATAFORMA DE STREAMING PARA VIDEOAULAS**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Daniel Facciolo Pires

**FRANCA/SP**  
**2024**

# **EDUSTREAM: Uma Plataforma de Streaming para Videoaulas**

Guilherme Hakime Andrade

## **Resumo**

O trabalho tem como objetivo o desenvolvimento de um protótipo funcional de uma plataforma de streaming de videoaulas, denominada EDUSTREAM, voltada para professores e alunos, permitindo o upload e visualização de conteúdos educacionais em forma de vídeo. O sistema é projetado utilizando técnicas, metodologias e linguagens de modelagem de engenharia de software: TAP (Termo de Abertura de Projeto), EAP (Estrutura Analítica do Projeto), SWOT (*Strengths, Weaknesses, Opportunities e Threats*), BPMN (*Business Process Model and Notation*), e UML (*Unified Modeling Language*). O protótipo foi desenvolvido utilizando a biblioteca ReactJS e a linguagem de folha de estilo CSS, oferecendo uma interface responsiva e amigável tanto para dispositivos móveis quanto para desktop. Este projeto busca atender à crescente demanda por plataformas educacionais online, principalmente em um contexto de ensino a distância, facilitando o acesso à educação de qualidade, além de expandir conhecimento para qualquer lugar.

## **Abstract**

*The objective of this work is to develop a functional prototype of a video lecture streaming platform, named EDUSTREAM, aimed at teachers and students, enabling the upload and viewing of educational content in video format. The system is designed using software engineering techniques, methodologies, and modeling languages: Project Charter (TAP), Work Breakdown Structure (WBS), SWOT (Strengths, Weaknesses, Opportunities, and Threats), BPMN (Business Process Model and Notation), and UML (Unified Modeling Language). The prototype was developed using the ReactJS library and CSS styling language, providing a responsive and user-friendly interface for both mobile and desktop devices. This project seeks to meet the growing demand for online educational*

*platforms, particularly in a distance learning context, facilitating access to quality education and significantly expanding knowledge to any location.*

## **1. Introdução**

O desenvolvimento de plataformas de streaming e ensino a distância tem ganhado relevância nos últimos anos, principalmente devido ao grande aumento da demanda por soluções tecnológicas que viabilizem o acesso remoto à educação. Este projeto aborda a criação de uma plataforma de *streaming* de videoaulas, denominada EDUSTREAM, com o intuito de fornecer um ambiente de fácil utilização, seguro e interativo para professores e alunos. A questão problema central deste trabalho é: como os métodos e técnicas da engenharia de software, bem como as tecnologias de programação web, podem apoiar o desenvolvimento de uma solução para acesso a conteúdos educacionais de forma interativa para usuários web?

### **1.1 Termo de Abertura do Projeto (TAP)**

O Termo de Abertura do Projeto (TAP) é um documento essencial que formaliza o início de um projeto. Ele define o propósito, objetivos e requisitos fundamentais, além de descrever o escopo e a equipe envolvida. O TAP tem como principal função obter a aprovação formal dos stakeholders e garantir que todos estejam alinhados com o propósito e as expectativas do projeto.

Segundo o Project Management Institute (PMI, 2017), o TAP permite a definição clara de responsabilidades e é utilizado para autorizar a utilização de recursos organizacionais no projeto (PMI, 2017). Sem ele, um projeto pode perder direção, já que oferece uma visão detalhada dos objetivos e das condições que precisam ser atendidas. Além disso, o TAP contribui para a gestão de riscos, já que possibilita a identificação de possíveis obstáculos e facilita o monitoramento de desempenho ao longo da execução (Vargas, 2014).

### **Termo de Abertura do Projeto (TAP)**

Nome do Projeto: Plataforma de Streaming de Videoaulas

Gerente do Projeto: Guilherme Hakime Andrade

Data de Início: 10/06/2024

### **Objetivo do Projeto:**

O objetivo deste projeto é desenvolver um protótipo funcional de uma plataforma de streaming de videoaulas voltada para professores e alunos, que permita o upload, download e gerenciamento e visualização de conteúdos educacionais. A plataforma será desenvolvida com tecnologias modernas como React, Tailwind CSS, garantindo uma interface responsiva e amigável para dispositivos móveis e desktops.

### **Justificativa:**

O crescimento do ensino a distância impulsionou a demanda por plataformas de educação online. A proposta deste projeto visa atender a essa necessidade, facilitando o acesso de alunos a videoaulas de alta qualidade.

### **Escopo:**

O escopo deste projeto inclui:

- Desenvolvimento da interface web responsiva para upload e visualização de videoaulas;
- Suporte a dispositivos móveis e desktops;
- Personalização de perfis de usuários (professores e alunos).

### **Cronograma do Projeto:**

O projeto terá início em julho de 2024, e será finalizado em novembro de 2024. Será realizado por partes dividido em semanas, para facilitar o monitoramento e garantir que os prazos sejam cumpridos

### **Orçamento Inicial:**

O projeto tem custo zero, pois será desenvolvido por um aluno para fins educativos.

### **Stakeholders:**

- Guilherme Hakime Andrade (Gerente do Projeto)
- Professores (Usuários da plataforma)
- Alunos (Usuários da plataforma)

### **Critérios de Sucesso:**

- Plataforma funcional e sem erros críticos;
- Interface amigável e intuitiva;
- Capacidade de upload e streaming de videoaulas de maneira fluida;
- Satisfação dos usuários (professores e alunos).

## **2. Estrutura Analítica do Projeto (EAP)**

A Estrutura Analítica do Projeto (EAP) organiza as atividades do projeto em uma estrutura hierárquica, facilitando o gerenciamento e controle das etapas. De acordo com Prado (2014), a EAP permite dividir o projeto em partes menores e mais gerenciáveis, proporcionando clareza no acompanhamento de prazos, custos e entregas.

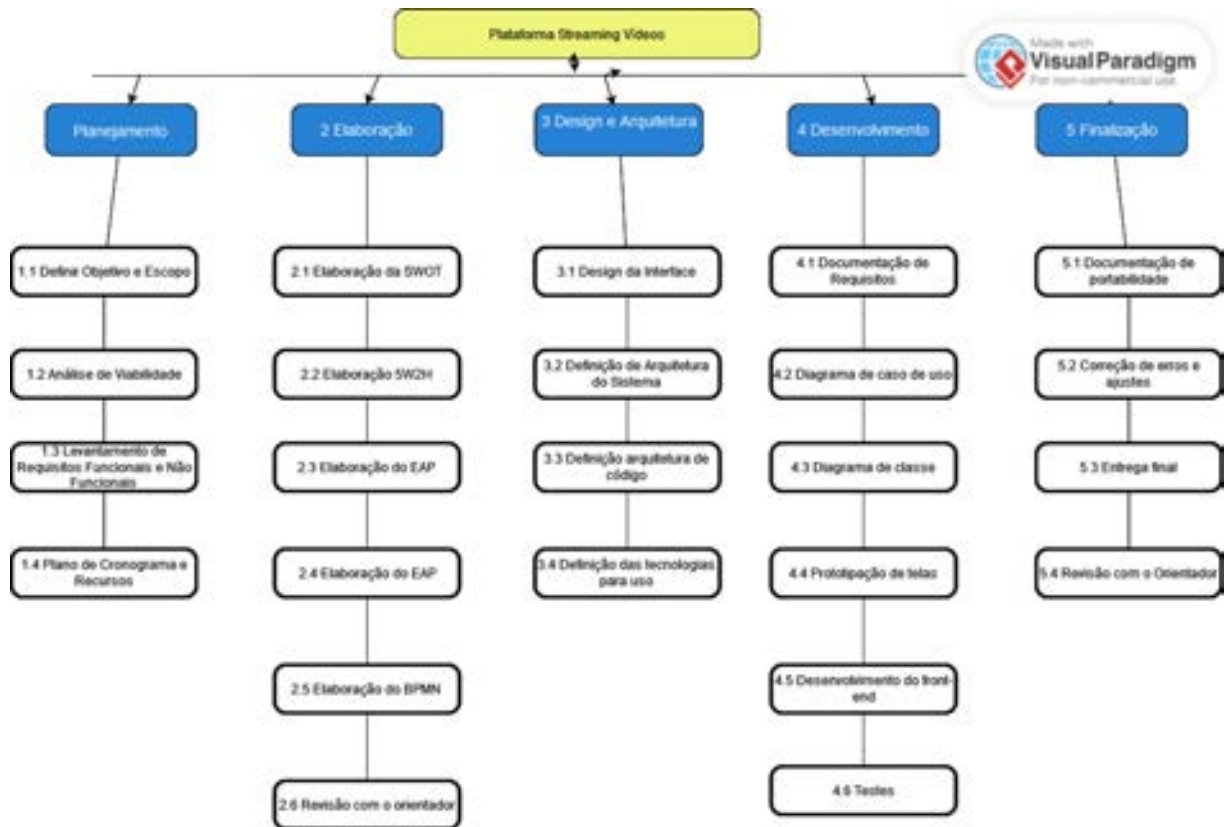
### **2.1. EAP da Plataforma de Streaming de Videoaulas**

No **Quadro 1 – Estrutura Analítica do Projeto (EAP)**, são apresentados cinco principais níveis do projeto: planejamento, elaboração, design e arquitetura, desenvolvimento e finalização.

A Figura 1 apresenta a (EAP) do desenvolvimento da plataforma de streaming de videoaulas. Nela, o projeto é subdividido em cinco níveis principais que correspondem às etapas essenciais para a criação e entrega da plataforma:

Cada nível detalha as atividades necessárias para a criação da plataforma, incluindo definição de requisitos, desenvolvimento técnico e testes.

Figura 1 - Estrutura Analítica do Projeto para a Plataforma de Streaming de Videoaulas



### 3 Fundamentação Teórica

Aqui serão apresentados os conceitos e fundamentos teóricos necessários para o desenvolvimento da plataforma de streaming de videoaulas, abordando o contexto do ensino a distância, os processos de automação para transmissão de vídeo, e as principais tecnologias para construção da plataforma.

#### 3.1 Ensino a Distância e Plataformas Digitais

Com o crescimento do ensino a distância (EAD), plataformas digitais se tornaram indispensáveis, possibilitando o acesso ao conhecimento a partir de qualquer lugar. De acordo com Ribeiro et al. (2021), o EAD promove a democratização do ensino, ampliando seu alcance e fornecendo flexibilidade para alunos e

professores. Assim percebemos que uma plataforma de streaming de videoaulas facilita a disseminação de conteúdos de forma prática e interativa ajudando qualquer pessoa em qualquer localização, com diferencial de uma plataforma simples e intuitiva.

## 3.2 Tecnologias para Desenvolvimento de Plataformas

A construção de uma plataforma de streaming requer o uso de tecnologias que garantam uma interface responsiva e intuitiva, além de uma boa experiência do usuário.

### 3.2.1 Arquitetura de Software

Para o projeto, utiliza-se uma arquitetura de duas camadas principais: **frontend** e **backend**. Essa arquitetura simples permite organizar a aplicação em componentes independentes, onde o frontend cuida da interface visual e da interação com o usuário, e o backend trata da lógica de negócio e das interações com o sistema de arquivos para armazenamento de vídeos. Segundo Clements e Kazman (2003), uma arquitetura modular e bem organizada facilita a escalabilidade e manutenção.

## 3.3 Processos Operacionais e Automação

Os processos operacionais são centrados no usuário e incluem atividades como autenticação, upload de vídeos e interação com o conteúdo, que buscam melhorar a experiência do usuário. Cada ação do usuário na interface dispara um processo de validação e feedback do sistema, garantindo uma experiência fluida. A automação desses processos, como a verificação das credenciais de acesso e a atualização do conteúdo de vídeo, contribui para a funcionalidade e segurança da plataforma.

## 3.4 Ferramentas e Frameworks Utilizados

Foram escolhidas ferramentas modernas para garantir eficiência, robustez e um design responsivo no desenvolvimento da plataforma.

### 3.4.1 Frontend: React e Tailwind CSS

No frontend, o React foi escolhido por sua flexibilidade e pela forma dinâmica como lida com interfaces modernas e responsivas (React Documentation, 2023). A utilização do Tailwind CSS permite criar um design visual agradável e adaptável a diferentes dispositivos, tornando a experiência do usuário mais agradável.

### 3.4.2 Backend: Next.js

No backend, o Next.js foi escolhido por sua capacidade de gerenciar rotas e integrar-se rapidamente ao React, facilitando o desenvolvimento simultâneo do frontend e backend. Suas funcionalidades, como o roteamento dinâmico e a renderização no lado do servidor, melhoram significativamente a performance da plataforma (Next.js Documentation, 2023).

## 4. Viabilidade do Projeto

### Blocos do BMC (Business Model Canvas):

1. **Segmento de Clientes:** professores e alunos do ensino a distância.
2. **Proposta de Valor:** facilitar o acesso a conteúdos educacionais de qualidade, oferecendo uma plataforma intuitiva e segura.
3. **Canais:** acesso via web e dispositivos móveis.
4. **Relacionamento:** personalização do conteúdo e suporte técnico.
5. **Recursos Chave:** infraestrutura de servidores, equipe de desenvolvimento.
6. **Atividades Chave:** desenvolvimento de novas funcionalidades, suporte e manutenção.
7. **Parcerias Chave:** instituições educacionais e provedores de infraestrutura.
8. **Estrutura de Custos:** custos com servidores, equipe de TI, marketing.

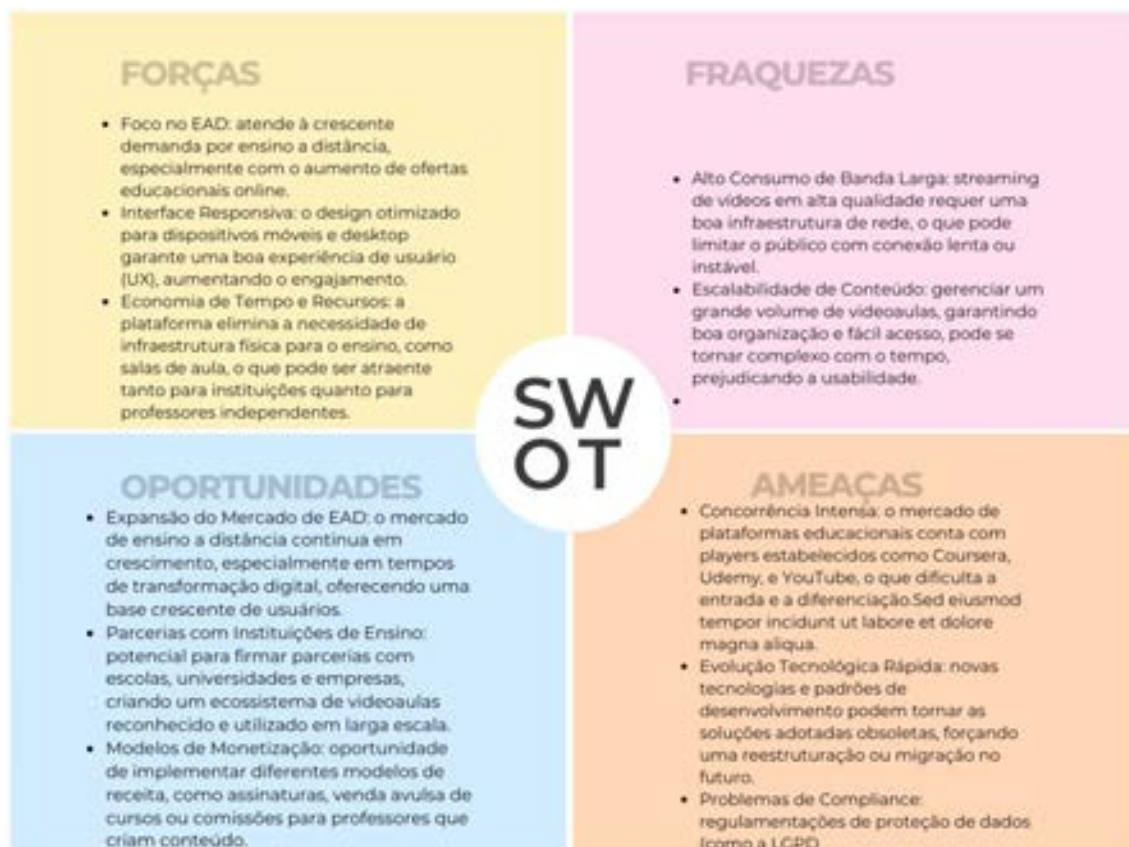
## 5. Matriz SWOT



A análise SWOT avalia os principais fatores internos e externos que podem influenciar o projeto. Ela é dividida em quatro áreas: forças e fraquezas (aspectos internos), oportunidades e ameaças (fatores externos), permitindo uma visão ampla dos elementos que podem impactar o sucesso da plataforma. A matriz SWOT é uma ferramenta estratégica amplamente utilizada para identificar as principais variáveis que afetam projetos, auxiliando no planejamento e na tomada de decisões (Chiavenato, 2014).

Na figura 2 – Matriz SWOT da Plataforma de Streaming de Videoaulas, pode ter uma visão detalhada dos pontos fortes e fracos, bem como das oportunidades e ameaças. Esta análise visa identificar onde a plataforma pode se destacar e quais desafios pode enfrentar ao longo de seu desenvolvimento.

Figura 2 – Matriz SWOT da Plataforma de Streaming de Videoaulas



## 6. Elicitação e Especificação dos Requisitos

A elicitação de requisitos é uma etapa essencial no desenvolvimento de software, pois permite identificar e compreender as necessidades e expectativas dos usuários e stakeholders. De acordo com Sommerville (2011), a elicitação é fundamental para garantir que o sistema seja desenvolvido conforme as demandas reais dos usuários, minimizando o risco de retrabalho e de funcionalidades inadequadas. Durante esse processo, utilizamos diversas técnicas para capturar e especificar os requisitos da plataforma de streaming de videoaulas, buscando uma solução robusta e alinhada aos objetivos do projeto.

### Técnicas de Elicitação Utilizadas

Para coletar informações detalhadas e precisas sobre os requisitos do sistema, aplicamos as seguintes técnicas de elicitação:

1. **Entrevistas:** realizadas com professores e alunos para identificar as funcionalidades essenciais, como upload de videoaulas, autenticação e interação com o conteúdo (comentários e avaliações).
2. **Visitas:** observamos plataformas de EAD para entender padrões de uso e pontos de melhoria que poderiam ser aplicados ao sistema.

As entrevistas e questionários possibilitaram coletar dados, garantindo uma visão ampla das necessidades dos usuários. As visitas complementaram essas informações, permitindo entender melhor o fluxo de uso e os principais pontos de atenção para os usuários. O questionário aplicado nas entrevistas está disponível no Apêndice A

### Especificação dos Requisitos

Os requisitos foram organizados em histórias de usuários para facilitar a compreensão e o desenvolvimento do sistema, destacando as funcionalidades principais da plataforma. Cada história descreve uma funcionalidade específica do ponto de vista do usuário:

- **História de Usuário 1:** "Como aluno, quero poder acessar a plataforma e visualizar as videoaulas disponíveis para estudar os conteúdos no meu ritmo."
- **História de Usuário 2:** "Como professor, quero ter a opção de fazer upload de videoaulas para disponibilizar meus conteúdos para os alunos."
- **História de Usuário 3:** "Como usuário autenticado, desejo comentar nas videoaulas para interagir com os professores e outros alunos."

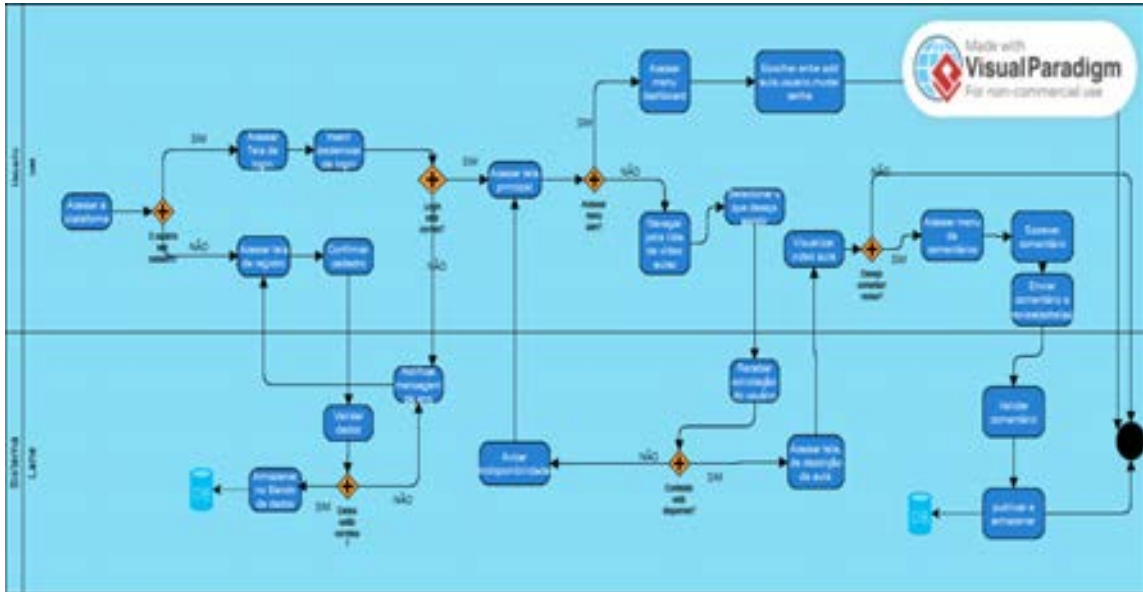
Cada um desses requisitos foi especificado com base nas respostas obtidas durante as questões das entrevistas feitas com alunos e professores. Detalhes adicionais sobre as perguntas e respostas coletadas podem ser encontrados no **Apêndice** ao final deste documento.

## 6.2 BUSINESS PROCESS MODEL AND NOTATION (BPMN).

O BPMN (*Business Process Model and Notation*) é uma notação gráfica que utiliza ícones para representar o fluxo de processos, permitindo um mapeamento claro e estruturado das atividades. Sua importância reside em padronizar a visualização dos processos de negócios, facilitando a análise, melhoria e automação das operações. O BPMN também contribui para a otimização da comunicação entre equipes e departamentos, sendo uma notação amplamente reconhecida e utilizada em diferentes setores (OMG, 2011).

Na Figura 3 – BPMN da Plataforma de Streaming de Videoaulas, são apresentados os processos principais de cada agente no sistema, representados por meio de raias distintas para o usuário e o sistema. Esse diagrama facilita a compreensão das interações e atividades de cada agente, permitindo que se identifiquem as responsabilidades e os fluxos de ação envolvidos na plataforma.

Figura 3 – BPMN da Plataforma de Streaming de Videoaulas



## Requisitos Funcionais

Segundo Pressman (2011), requisitos funcionais descrevem as funcionalidades ou serviços que o sistema deve oferecer, especificando o comportamento que o software precisa ter para atender às expectativas do usuário. Esses requisitos detalham as operações que o sistema deve realizar, incluindo processamento, manipulação de dados e interações com o usuário.

A seguir, apresentamos a **Tabela 1 – Requisitos Funcionais do Sistema de Streaming de Videoaulas**, que delinea as principais funcionalidades que a plataforma deve disponibilizar. Cada requisito é descrito de forma a evidenciar as operações esperadas, contribuindo para uma compreensão clara do que o sistema deve ser capaz de realizar para atender às necessidades dos usuários.

Tabela 1. Requisitos funcionais

<b>Código</b>	<b>Requisito Funcional</b>	<b>Categoria</b>	<b>Prioridade</b>	<b>Descrição</b>
RF001	Acessar Plataforma	Evidente	Altíssima	O sistema deve permitir que o usuário acesse a plataforma inicial.
RF002	Tela de Login e Registro	Evidente	Altíssima	O sistema deve apresentar uma tela para o usuário fazer login ou se registrar.
RF003	Inserção e Armazenamento de Credenciais	Oculto	Alta	O sistema deve permitir que o usuário insira suas credenciais e armazene os dados de login.
RF004	Validação de Credenciais	Oculto	Altíssima	O sistema deve validar as credenciais fornecidas antes de autorizar o acesso ao sistema.
RF005	Acesso à Tela Principal	Evidente	Altíssima	O sistema deve direcionar o usuário autenticado para a tela principal da plataforma.
RF006	Navegação pela Lista de Videoaulas	Evidente	Alta	O sistema deve permitir que o usuário navegue pela lista de videoaulas disponíveis.
RF007	Seleção de Videoaula	Evidente	Altíssima	O sistema deve permitir que o usuário selecione uma videoaula que deseja assistir.
RF008	Visualização de Videoaula	Evidente	Altíssima	O sistema deve permitir que o usuário assista ao vídeo selecionado.
RF009	Aviso de Indisponibilidade	Oculto	Alta	O sistema deve informar o usuário caso a videoaula

<b>Código</b>	<b>Requisito Funcional</b>	<b>Categoria</b>	<b>Prioridade</b>	<b>Descrição</b>
				selecionada não esteja disponível.
RF010	Acesso ao Menu de Comentários e Avaliações	Evidente	Média	O sistema deve permitir que o usuário acesse o menu de comentários e avaliações.
RF011	Envio de Comentários e Avaliações	Evidente	Alta	O sistema deve permitir que o usuário envie comentários e avaliações de até 5 estrelas.
RF012	Acesso ao Painel Administrativo	Oculto	Altíssima	O sistema deve permitir que o usuário acesse o painel de administrador com opções de gerenciamento.
RF013	Adição de Videoaula no Painel Administrativo	Oculto	Alta	O sistema deve permitir que o usuário adicione uma nova videoaula pelo painel de administrador.
RF014	Remoção de Videoaula no Painel Administrativo	Oculto	Alta	O sistema deve permitir que o usuário remova uma videoaula pelo painel de administrador.
RF015	Adição de Disciplinas no Painel Administrativo	Oculto	Média	O sistema deve permitir que o usuário adicione novas disciplinas no painel de administrador.
RF016	Atualização de Dados do Usuário	Oculto	Média	O sistema deve permitir que o usuário atualize informações como senha no painel administrativo.

Código	Requisito Funcional	Categoria	Prioridade	Descrição
RF017	Armazenamento de Comentários e Avaliações	Oculto	Alta	O sistema deve armazenar os comentários e avaliações no banco de dados.

### 6.3 Requisitos Não Funcionais

De acordo com Sommerville (2011), **requisitos não funcionais** especificam critérios que podem ser usados para avaliar o desempenho de um sistema, em vez de descrever as funcionalidades que ele deve ter. Esses requisitos estão relacionados a aspectos como usabilidade, confiabilidade, desempenho e segurança, que afetam a maneira como o sistema opera, mas não diretamente as funcionalidades oferecidas.

Abaixo, são apresentados os requisitos não funcionais do sistema de streaming de videoaulas, organizados em um quadro que detalha cada aspecto relevante para o projeto.

Tabela 2 – Requisitos Não Funcionais do Sistema

Código	Requisito Não Funcional	Tipo	Prioridade	Descrição
RNF001	Tempo de Resposta Rápido	Obrigatório	Permanente	O sistema deve carregar o conteúdo das videoaulas em até 6 segundos para uma boa experiência de uso.
RNF002	Compatibilidade com Dispositivos	Obrigatório	Permanente	O sistema deve ser compatível com dispositivos móveis e desktops.
RNF003	Segurança de Dados	Obrigatório	Permanente	O sistema deve criptografar dados de login e senhas.

<b>Código</b>	<b>Requisito Não Funcional</b>	<b>Tipo</b>	<b>Prioridade</b>	<b>Descrição</b>
RNF004	Escalabilidade	Obrigatório	Permanente	O sistema deve suportar aumento de usuários e de conteúdo sem comprometer a performance.
RNF005	Alta Disponibilidade	Obrigatório	Permanente	O sistema deve garantir 99,9% de disponibilidade, minimizando quedas e indisponibilidades.
RNF006	Manutenibilidade	Desejável	Permanente	O sistema deve permitir fácil manutenção e atualização do código e da infraestrutura.
RNF007	Interface Intuitiva	Obrigatório	Permanente	O sistema deve ser fácil de usar e apresentar interface intuitiva para uma boa experiência do usuário.

#### 6.4 Regras de Negócio

Segundo Kotonya e Sommerville (1998), **regras de negócio** são diretrizes ou restrições que governam as operações de um sistema de software, baseadas em políticas organizacionais, leis ou normas. Elas refletem decisões e políticas fundamentais para o funcionamento da empresa e impactam diretamente como o sistema deve operar.

A importância das regras de negócio, conforme explica Pressman (2011), está em garantir que o sistema atenda às expectativas e necessidades do negócio, alinhando as operações do software aos objetivos e práticas da organização. Sem essas regras claramente definidas, o sistema pode gerar resultados inconsistentes.

Abaixo, apresentamos as regras de negócio do sistema de streaming de videoaulas, conforme o modelo de documentação estudado.



Tabela 3 – Regras de Negócio do Sistema

<b>Código</b>	<b>Regra de Negócio</b>	<b>Descrição</b>
RN001	Cadastro de Usuário	O usuário deve fornecer um nome, email e senha para se cadastrar. O email deve ser único.
RN002	Validação de Credenciais	O sistema deve validar as credenciais de login, permitindo acesso apenas a usuários com informações corretas.
RN003	Validação de Cadastro	Durante o cadastro, o sistema deve validar que todos os campos obrigatórios foram preenchidos corretamente e que a senha atende aos requisitos de segurança (mínimo de 8 caracteres, contendo letras e números).
RN004	Limite de Comentários	O usuário pode enviar no máximo um comentário por videoaula para evitar spam.
RN005	Validação de Comentários	O sistema deve validar os comentários enviados para garantir que não contenham conteúdo ofensivo ou impróprio.
RN006	Disponibilidade do Conteúdo	O sistema deve verificar a disponibilidade do conteúdo antes de permitir que o usuário inicie a transmissão da videoaula.
RN007	Armazenamento de Dados	Todas as informações de usuários, videoaulas e comentários devem ser armazenadas de forma segura no banco de dados.
RN008	Acesso Restrito a Conteúdo	Conteúdos sensíveis ou com restrição devem exigir autenticação ou permissões específicas para acesso.

<b>Código</b>	<b>Regra de Negócio</b>	<b>Descrição</b>
RN009	Notificação de Sucesso em Cadastro	Após o cadastro, o usuário deve receber uma notificação de sucesso e instruções para realizar o login.
RN010	Registro de Atividades	O sistema deve registrar todas as atividades dos usuários (como login, visualização de videoaulas e envio de comentários) para fins de análise e melhoria contínua.

## 6.5 Casos de Uso

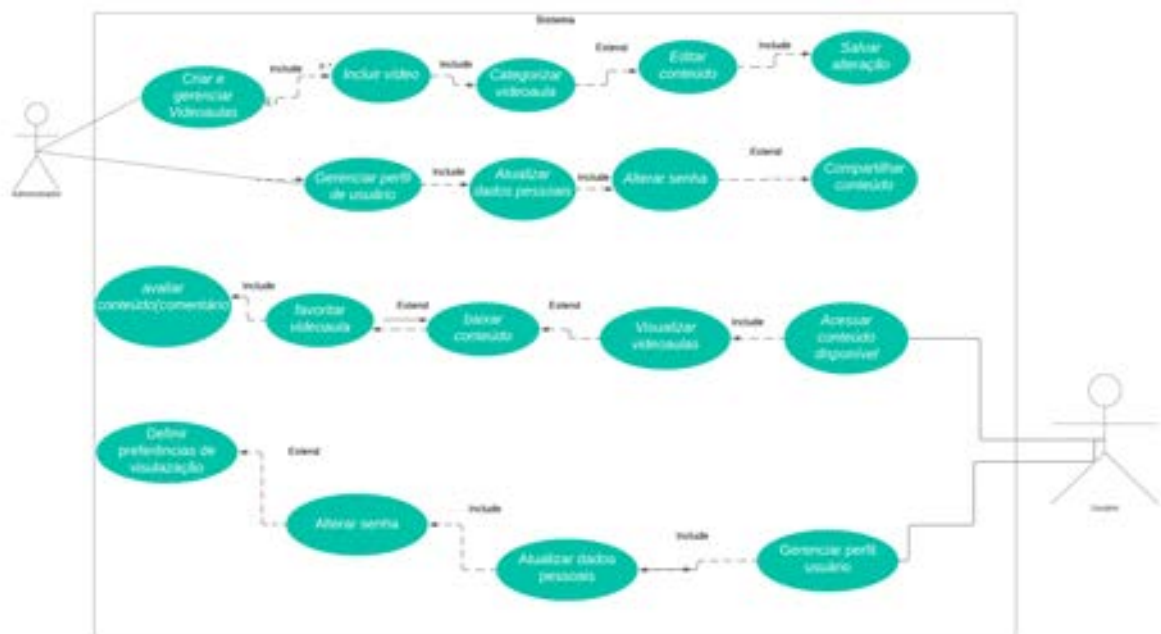
De acordo com Jacobson (1992), **casos de uso** são descrições das interações entre um sistema e seus usuários ou outros sistemas, com o objetivo de realizar uma tarefa específica. Eles são representações fundamentais no processo de modelagem de sistemas.

A importância dos casos de uso, como afirmam Booch, Rumbaugh e Jacobson (1999), está em fornecer uma visão clara e objetiva das funcionalidades que o sistema deve oferecer, facilitando a comunicação entre desenvolvedores, analistas e stakeholders.

**Diagrama de caso de uso:** no diagrama, temos dois atores principais: administrador/Professor e aluno. Ambos interagem com o sistema para realizar tarefas essenciais, como gestão de aulas, categorias e usuários (para o Administrador/Professor) e acesso a aulas e funcionalidades de feedback (para o Aluno). O sistema, por sua vez, automatiza o gerenciamento de dados e a geração de relatórios, oferecendo recomendações personalizadas e processando dados de feedback para melhorar a experiência dos usuários.

Conforme afirmado, "o diagrama de caso de uso facilita a visualização das interações entre os usuários e o sistema, permitindo uma compreensão clara das funcionalidades-chave e da estrutura funcional de um sistema" (Silva, 2024).

Figura 4– Diagrama de caso de uso da plataforma



include

**Tabela 4 – Caso de Uso: Cadastrar Usuários**

<b>Elemento</b>	<b>Detalhes</b>
<b>ID</b>	UC001
<b>Caso de Uso</b>	Cadastrar Usuário
<b>Descrição</b>	Este caso de uso descreve o processo de cadastro de um novo usuário no sistema.
<b>Ator Primário</b>	Usuário do sistema
<b>Pré-condição</b>	Nenhuma
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção de cadastro.</li> <li>2. O sistema carrega o formulário de cadastro.</li> <li>3. O sistema solicita informações como nome, e-mail, e senha.</li> <li>4. O usuário preenche os dados e confirma.</li> <li>5. O sistema valida os dados.</li> <li>6. O sistema cadastra o novo usuário.</li> </ol>
<b>Pós-condição</b>	O usuário é cadastrado com sucesso e pode acessar o sistema.
<b>Cenário Alternativo</b>	<p>4a – O usuário insere dados inválidos.</p> <p>4a.1 O sistema exibe uma mensagem de erro e solicita a correção dos dados.</p>

**Tabela 5. Índice de Casos de Uso**

<b>ID</b>	<b>Nome do Caso de Uso</b>	<b>Descrição</b>
<b>UC001</b>	Cadastrar Usuário	Permite que novos usuários se cadastrem no sistema, inserindo dados pessoais.
<b>UC002</b>	Fazer Login	Permite que usuários previamente cadastrados façam login para acessar suas contas.
<b>UC003</b>	Visualizar Videoaulas	Alunos podem visualizar videoaulas disponíveis na plataforma.
<b>UC004</b>	Comentar em Videoaulas	Alunos podem adicionar comentários nas videoaulas assistidas.
<b>UC005</b>	Gerenciar Perfil	Usuários podem editar e atualizar seus dados pessoais.

### **Especificação de Casos de Uso**

Abaixo segue a especificação detalhada de cada caso de uso, conforme o modelo de documentação estudado nas aulas de Engenharia de Software.

Tabela 6 – Caso de Uso: fazer Login

<b>Elemento</b>	<b>Detalhes</b>
<b>ID</b>	UC002
<b>Caso de Uso</b>	Fazer Login
<b>Descrição</b>	Este caso de uso descreve o processo de login de um usuário já cadastrado.
<b>Ator Primário</b>	Usuário do sistema

<b>Elemento</b>	<b>Detalhes</b>
<b>Pré-condição</b>	O usuário deve estar previamente cadastrado.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção de login.</li> <li>2. O sistema carrega o formulário de login.</li> <li>3. O usuário insere e-mail e senha.</li> <li>4. O sistema valida as credenciais.</li> <li>5. O sistema autentica o usuário e concede acesso.</li> </ol>
<b>Pós-condição</b>	O usuário é autenticado e pode acessar suas funcionalidades.
<b>Cenário Alternativo</b>	<p>3a – O usuário insere credenciais incorretas.</p> <p>3a.1 O sistema exibe uma mensagem de erro e solicita a correção dos dados.</p>

Tabela 7 – Caso de Uso: Visualizar Videoaulas

<b>Elemento</b>	<b>Detalhes</b>
<b>ID</b>	UC003
<b>Caso de Uso</b>	Visualizar Videoaulas
<b>Descrição</b>	Este caso de uso descreve o processo pelo qual um aluno pode visualizar videoaulas.
<b>Ator Primário</b>	Aluno
<b>Pré-condição</b>	O aluno deve estar autenticado no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O aluno acessa a seção de videoaulas.</li> <li>2. O sistema exibe a lista de videoaulas disponíveis.</li> </ol>

<b>Elemento</b>	<b>Detalhes</b>
	3. O aluno seleciona uma videoaula. 4. O sistema carrega o player de vídeo e inicia a reprodução.
<b>Pós-condição</b>	O aluno visualiza a videoaula selecionada.
<b>Cenário Alternativo</b>	3a – O aluno seleciona uma videoaula que não está disponível. 3a.1 O sistema exibe uma mensagem informando que a videoaula não está disponível.

Tabela 8 – Caso de Uso: Comentar em Videoaulas

<b>Elemento</b>	<b>Detalhes</b>
<b>ID</b>	UC004
<b>Caso de Uso</b>	Comentar em Videoaulas
<b>Descrição</b>	Este caso de uso descreve o processo pelo qual um aluno pode comentar em uma videoaula assistida.
<b>Ator Primário</b>	Aluno
<b>Pré-condição</b>	O aluno deve ter assistido a pelo menos 50% da videoaula.
<b>Cenário Principal</b>	1. O aluno acessa a videoaula assistida. 2. O sistema exibe a opção de comentário. 3. O aluno digita seu comentário e clica em "Enviar". 4. O sistema valida o comentário e o publica. 5. O sistema atualiza a lista de comentários.
<b>Pós-condição</b>	O comentário do aluno é exibido na seção de comentários da videoaula.

<b>Elemento</b>	<b>Detalhes</b>
<b>Cenário Alternativo</b>	3a – O aluno insere um comentário que contém conteúdo impróprio. 3a.1 O sistema exibe uma mensagem de erro informando que o conteúdo é inaceitável.

Tabela 9 – Caso de Uso: Gerenciar Perfil

<b>Elemento</b>	<b>Detalhes</b>
<b>ID</b>	UC005
<b>Caso de Uso</b>	Gerenciar Perfil
<b>Descrição</b>	Este caso de uso descreve o processo pelo qual um usuário pode editar suas informações pessoais.
<b>Ator Primário</b>	Usuário do sistema
<b>Pré-condição</b>	O usuário deve estar autenticado no sistema.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção de "Gerenciar Perfil".</li> <li>2. O sistema carrega as informações do perfil.</li> <li>3. O usuário altera os dados desejados (nome, e-mail, senha).</li> <li>4. O sistema valida as informações.</li> <li>5. O usuário confirma as alterações.</li> <li>6. O sistema atualiza as informações no banco de dados.</li> </ol>
<b>Pós-condição</b>	As informações do perfil do usuário são atualizadas com sucesso.



Elemento	Detalhes
<b>Cenário Alternativo</b>	4a – O usuário insere um e-mail inválido. 4a.1 O sistema exibe uma mensagem de erro informando que o e-mail não é válido.

## 6.7 Diagrama de Classe

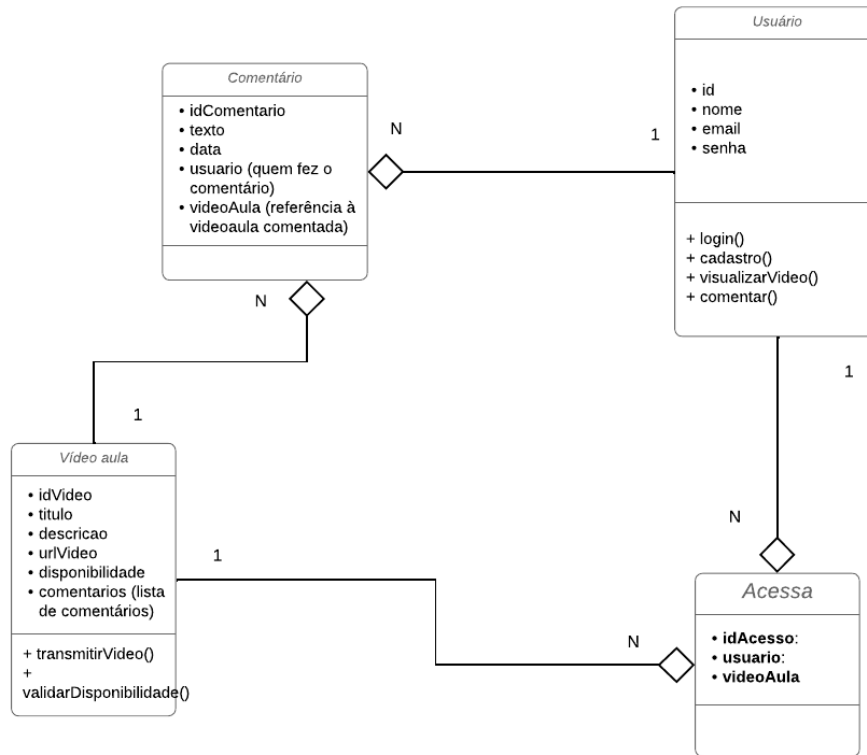
Um diagrama de classe é uma representação visual que ilustra classes, atributos, métodos e os relacionamentos entre eles em um sistema orientado a objetos, sendo uma parte essencial da UML (Unified Modeling Language) (Larman, 2004). Ele fornece uma visão estruturada do sistema, permitindo identificar como as diferentes partes interagem e se relacionam entre si.

A seguir, apresentamos o **Diagrama de Classe** (Figura 5), que detalha as principais classes do sistema de streaming de videoaulas, incluindo seus atributos e métodos. Este diagrama ajuda a entender a estrutura do software, facilitando o desenvolvimento e a manutenção do sistema, nele será mostrado a interação de vídeo-aulas, usuários, tela de comentários.

*Figura 5 – Diagrama de Classe da Plataforma de Streaming de Videoaulas*

## Diagrama de classe UML

Gui Land | November 5, 2024



## 7. Desenvolvimento

O desenvolvimento de produtos digitais envolve a aplicação de princípios que garantem funcionalidade, usabilidade e uma boa experiência ao usuário.

### 7.1. Conceitos e Fundamentos sobre UX e IHC

**Experiência do Usuário (UX)** refere-se ao conjunto de sentimentos e percepções que um usuário tem ao interagir com um produto digital. A **Interação Humano-Computador (IHC)** estuda como as pessoas interagem com computadores e sistemas, visando melhorar a usabilidade e a eficiência das interfaces.

A importância de UX e IHC no desenvolvimento de produtos digitais é fundamental, pois uma boa experiência do usuário pode aumentar a retenção, reduzir a taxa de abandono e melhorar a imagem da marca.

## 7.2. Prototipagem

A prototipagem é o processo de criar representações de um produto para testar e validar ideias antes da implementação final.

### 7.2.1. Benefícios de investir em prototipagem:

**Permite identificar problemas e realizar ajustes antes da implementação.** A prototipagem possibilita a visualização de funcionalidades e fluxos de interação do sistema, o que ajuda a detectar problemas potenciais antes que o desenvolvimento final comece (Schmidt, 2008).

**Economiza tempo e recursos a longo prazo.** Investir na criação de protótipos pode reduzir significativamente o tempo e os custos associados a mudanças posteriores, pois as falhas são detectadas precocemente, evitando retrabalho e desperdício de recursos (Brown, 2009).

**Facilita a comunicação entre equipes e stakeholders.** Os protótipos funcionais servem como uma ferramenta visual que melhora a compreensão mútua entre desenvolvedores, designers e partes interessadas, promovendo um diálogo mais eficaz sobre expectativas e requisitos (Snyder, 2003).

## 7.3. Processo de Desenvolvimento do Protótipo de Interface

O desenvolvimento de um protótipo de interface é uma etapa importante no design de produtos digitais, pois permite visualizar e testar as interações do usuário antes da implementação final. Um protótipo serve como um modelo preliminar que representa a estrutura e as funcionalidades do produto, facilitando a identificação de problemas e a realização de ajustes antes da produção. Essa abordagem é essencial para garantir que as necessidades e expectativas dos usuários sejam atendidas, economizando tempo e recursos a longo prazo (Rosenfeld et al., 2015).

### 7.3.1. Briefing:

Levantamento de informações sobre o projeto, objetivos e requisitos dos usuários.

### **7.3.2. Plano de Ação 5W1H – DO PROBLEMA:**

What: O que é o problema? (falta de acesso fácil a videoaulas).

Why: Por que isso é um problema? (dificuldade em encontrar conteúdos educativos).

When: Quando o problema ocorre? (principalmente em contextos de ensino a distância).

Where: Onde o problema é mais evidente? (plataformas de ensino online).

Who: Quem é afetado? (alunos e professores).

How: Como o problema pode ser resolvido? (criando uma plataforma intuitiva).

### **7.3.3. Personas:**

Criação de perfis fictícios representando os usuários-alvo (ex: professor experiente, aluno do ensino médio). Isso ajuda a entender suas necessidades e expectativas.

### **7.3.4. Suposições - Hipóteses – Validação:**

Estabelecer suposições sobre o comportamento do usuário, criar hipóteses a partir delas e validar com testes e feedback.

### **7.3.5. Benchmark:**

Análise de concorrentes e plataformas similares identificando assim melhorias, e oportunidades.

### **7.3.6. Mapa de Jornada de Usuários:**

Criação de um mapa que ilustra a experiência do usuário ao interagir com a plataforma, destacando pontos de dor e oportunidades.

#### **7.3.7. Rabiscoframe (Canvas 4-Cantos):**

Utilização de um canvas para brainstorm e rascunhos, onde cada canto representa um elemento importante (ex: objetivo, público-alvo, funcionalidades, diferenciais).

#### **7.3.8. Wireframe:**

Desenvolvimento de um layout básico que apresenta a estrutura da interface, sem detalhes visuais.

#### **7.3.9. Styleguide:**

Criação de um guia de estilo que define cores, tipografia, ícones e componentes, garantindo consistência visual em toda a plataforma, melhorando assim o UX UI.

#### **7.3.10. Protótipo de Alta Resolução – Navegável:**

Desenvolvimento de um protótipo interativo que simula a experiência final, permitindo que usuários testem a interface e forneçam feedback.

#### **7.3.11. Avaliação Heurística:**

Realização de uma avaliação da interface com base em princípios de usabilidade (heurísticas), identificando problemas e áreas de melhoria.

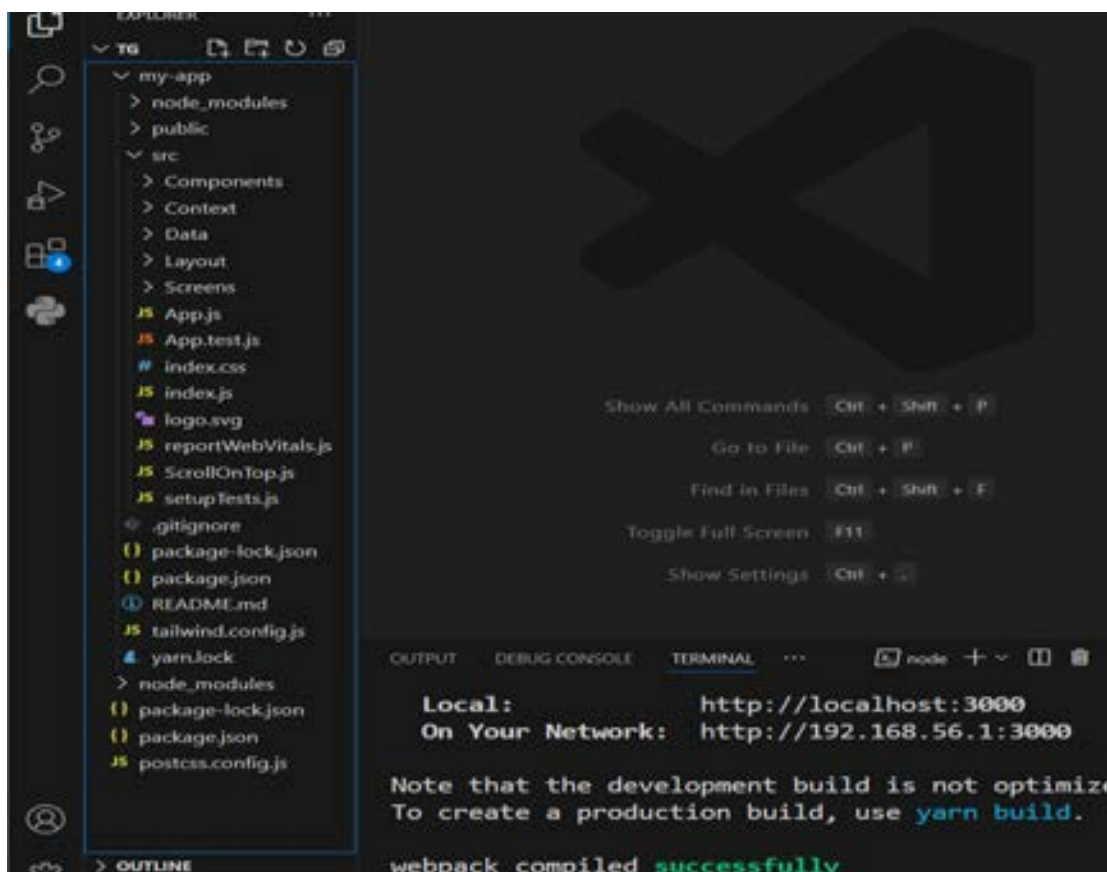
## **8. Telas e código da plataforma EDUSTREAM**

### **8.1. Arquitetura do projeto**

Este projeto de plataforma de streaming de videoaulas é estruturado em módulos reutilizáveis para páginas, componentes e modais. Usa Layout e SideBar para organizar a navegação e páginas principais, como Home, Login e Aulas, além de uma área administrativa para gerenciar conteúdo (aulas e categorias). Os dados são centralizados em objetos (ex: ClassesData), permitindo fácil controle e renderização. A estilização, provavelmente com Tailwind CSS, assegura uma interface responsiva. A arquitetura é modular, facilitando a manutenção e a expansão da plataforma.

Segue a Figura 6, mostrando a organização das pastas do projeto, tendo como raiz a pasta my-app, contendo suas dependências necessárias, também a pasta src contendo componentes, dados, telas e layout, e a pasta public onde estão as imagens.

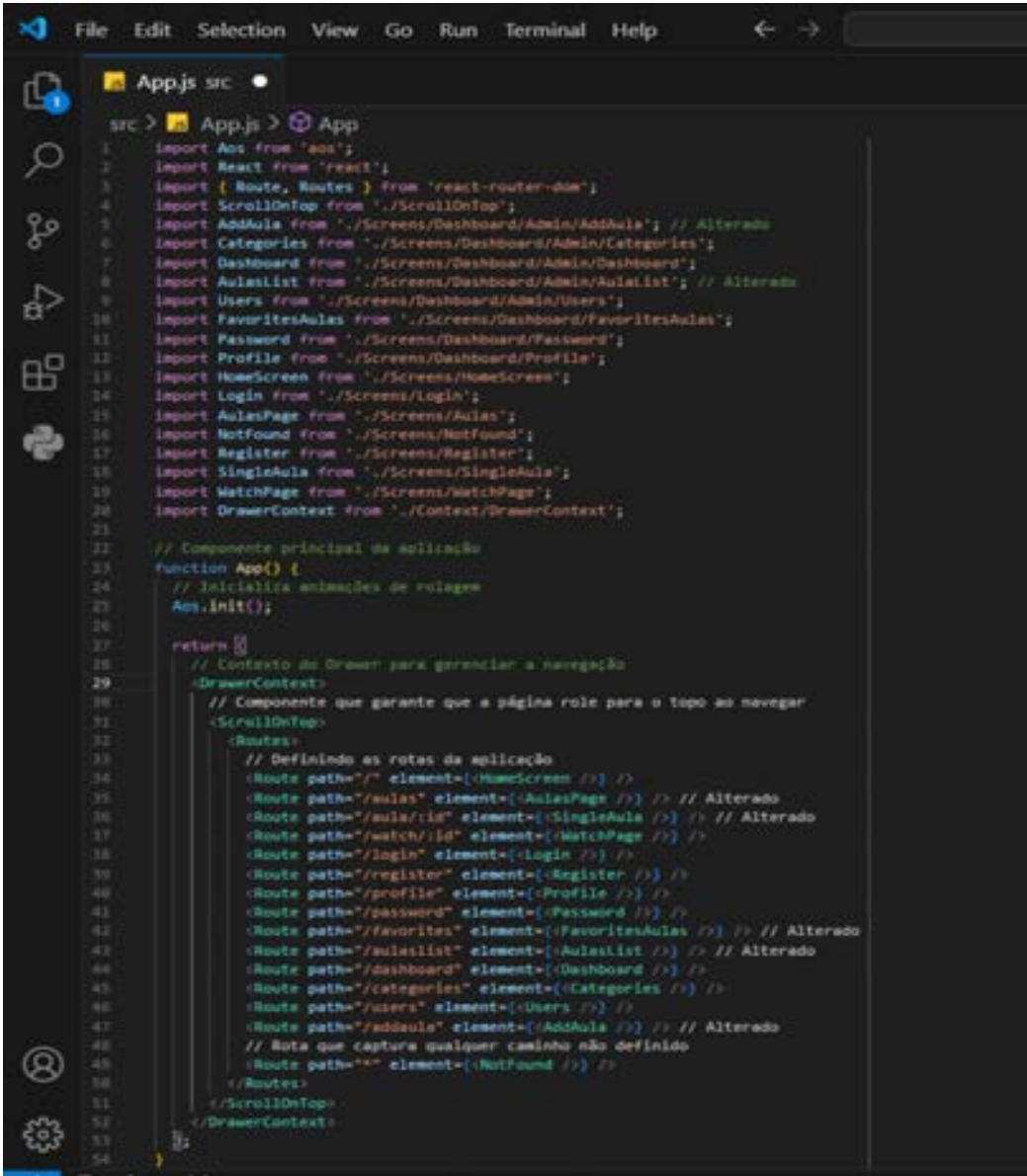
Figura 6 – Organização das pastas do projeto



A Figura 7 abaixo representa o componente principal de uma aplicação React para um sistema de streaming de aulas. Ele utiliza o React Router para gerenciar

as rotas da aplicação, permitindo a navegação entre diferentes telas, como a página inicial, login, registro, perfil e a lista de aulas. O DrawerContext facilita a comunicação entre os componentes, enquanto o ScrollOnTop garante que a página role para o topo ao mudar de rota. O uso da biblioteca AOS (Animate On Scroll) proporciona animações durante a rolagem, melhorando a experiência do usuário.

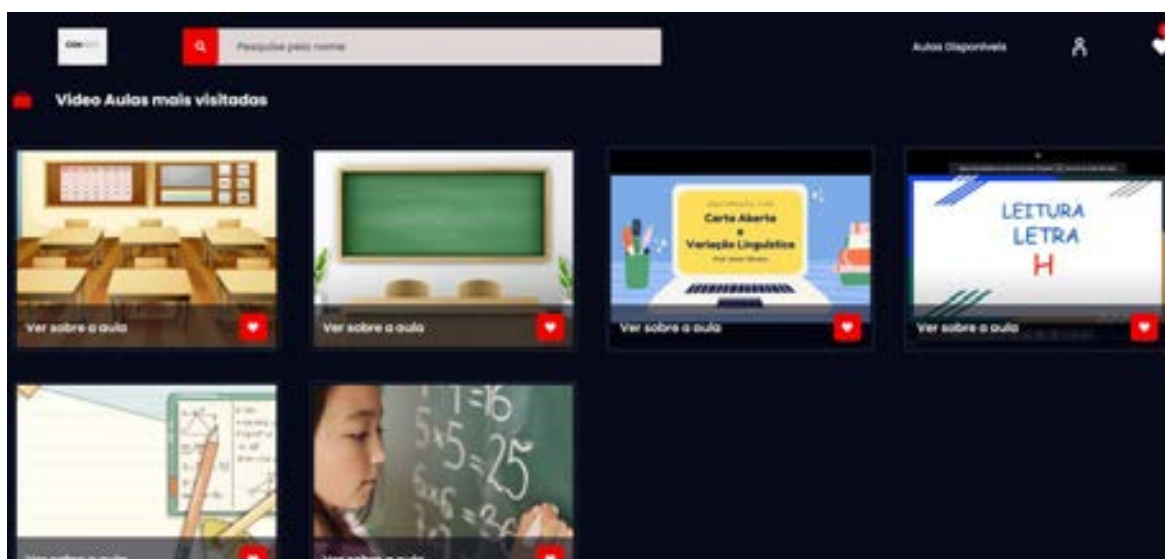
Figura 7 - Estrutura de Rotas da Aplicação de Streaming de Videoaulas



```
1 import Aos from 'aos';
2 import React from 'react';
3 import { Route, Routes } from 'react-router-dom';
4 import ScrollOnTop from './ScrollOnTop';
5 import AddAula from './Screens/Dashboard/Admin/AddAula'; // Alterado
6 import Categories from './Screens/Dashboard/Admin/Categories';
7 import Dashboard from './Screens/Dashboard/Admin/Dashboard';
8 import AulasList from './Screens/Dashboard/Admin/AulasList'; // Alterado
9 import Users from './Screens/Dashboard/Admin/Users';
10 import FavoritesAulas from './Screens/Dashboard/FavoritesAulas';
11 import Password from './Screens/Dashboard/Password';
12 import Profile from './Screens/Dashboard/Profile';
13 import HomeScreen from './Screens/HomeScreen';
14 import Login from './Screens/Login';
15 import AulasPage from './Screens/Aulas';
16 import NotFound from './Screens/NotFound';
17 import Register from './Screens/Register';
18 import SingleAula from './Screens/SingleAula';
19 import WatchPage from './Screens/WatchPage';
20 import DrawerContext from './Context/DrawerContext';
21
22 // Componente principal da aplicação
23 function App() {
24   // Inicializa animações de rolagem
25   Aos.init();
26
27   return (
28     // Contexto de Drawer para gerenciar a navegação
29     <DrawerContext>
30       // Componente que garante que a página role para o topo ao navegar
31       <ScrollOnTop>
32         <Routes>
33           // Definindo as rotas da aplicação
34           <Route path="/" element={<HomeScreen />} />
35           <Route path="/aulas" element={<AulasPage />} /> // Alterado
36           <Route path="/aula/:id" element={<SingleAula />} /> // Alterado
37           <Route path="/watch/:id" element={<WatchPage />} />
38           <Route path="/login" element={<Login />} />
39           <Route path="/register" element={<Register />} />
40           <Route path="/profile" element={<Profile />} />
41           <Route path="/password" element={<Password />} />
42           <Route path="/favorites" element={<FavoritesAulas />} /> // Alterado
43           <Route path="/aulaslist" element={<AulasList />} /> // Alterado
44           <Route path="/dashboard" element={<Dashboard />} />
45           <Route path="/categories" element={<Categories />} />
46           <Route path="/users" element={<Users />} />
47           <Route path="/addaula" element={<AddAula />} /> // Alterado
48           // Rota que captura qualquer caminho não definido
49           <Route path="*" element={<NotFound />} />
50         </Routes>
51       </ScrollOnTop>
52     </DrawerContext>
53   );
54 }
```

Na Figura 8 a Tela Home da plataforma de streaming de aulas apresenta um menu de pesquisa, uma barra de navegação (navbar) e uma lista das vídeo aulas mais acessadas. O menu de pesquisa facilita a busca por conteúdos, enquanto a barra de navegação proporciona acesso rápido a diferentes seções. A lista de vídeo aulas em destaque incentiva a exploração, tornando a experiência do usuário mais eficiente e envolvente.

Figura 8 - Tela Home da Plataforma de Streaming de Aulas



Na Figura 9 a Tela Home da plataforma apresenta um layout com o **Banner** (linha 6) para promoções, aulas mais populares (linha 5) através do componente **PopularAulas**, promoções adicionais (linha 7) e as aulas mais bem avaliadas (linha 8).

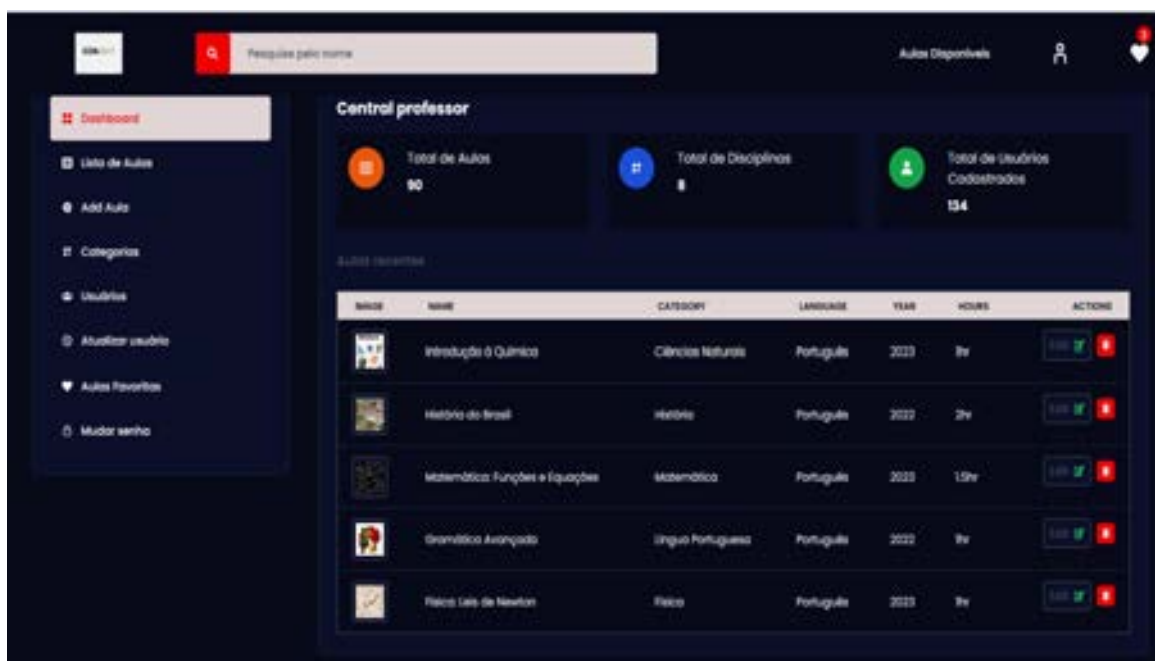
Figura 9 – Código da Tela Home da Plataforma de Streaming de Aulas



```
my-app > src > Screens > HomeScreen.js > default
1 import React from 'react';
2 // Importa o layout principal da aplicação
3 import layout from '../Layout/Layout';
4 // Importa os componentes da página inicial, incluindo as aulas populares, banner, promoções e as mais bem avaliadas
5 import PopularAulas from '../Components/Home/PopularAulas';
6 import Banner from '../Components/Home/Banner';
7 import Promos from '../Components/Home/Promos';
8 import TopRated from '../Components/Home/TopRated';
9
10 // Define o componente da tela inicial
11 function HomeScreen() {
12   return (
13     <Layout>
14       <div className="container mx-auto min-h-screen px-2 mb-6">
15         <Banner /> { /* Componente de banner da página inicial */}
16         <PopularAulas /> { /* Componente para exibir aulas populares */}
17         <Promos /> { /* Componente de promoções */}
18         <TopRated /> { /* Componente para exibir aulas mais bem avaliadas */}
19       </div>
20     </Layout>
21   );
22 }
23
24 // exporta o componente da tela inicial
25 export default HomeScreen;
```

Na Figura 10 segue a tela do painel administrativo da plataforma, que contém um menu lateral com diversas opções de navegação para o administrador.

Figura 10 – Tela de painel administrativo do sistema.



Este código da Figura 11 de componente define uma barra lateral de navegação com links para diferentes páginas, incluindo aulas, categorias, e configurações de usuário. Ele utiliza NavLink do React Router para navegação e estilos dinâmicos para destacar o link ativo.

Figura 11 – Código da Tela de painel administrativo do sistema.

```
app > src > Screens > Dashboard > Sidelbar.js > Sidellar > Sidelinks
import React from "react";
import { BsFillGridFill } from "react-icons/bs";
import { FaListAlt, FaUsers, FaHeart } from "react-icons/fa";
import { RiMovie2Fill, RiLockPasswordLine } from "react-icons/ri";
import { HiViewGridAdd } from "react-icons/hi";
import { FiSettings } from "react-icons/fi";
import Layout from "../../Layout/Layout";
import { NavLink } from "react-router-dom";

function SideBar({ children }) {
  const Sidelinks = [
    {
      name: "Dashboard",
      link: "/dashboard",
      icon: BsFillGridFill,
    },
    {
      name: "lista de Aulas",
      link: "/movieslist",
      icon: FaListAlt,
    },
    {
      name: "Add Aula",
      link: "/addmovie",
      icon: RiMovie2Fill,
    },
    {
      name: "Categorias",
      link: "/categories",
      icon: HiViewGridAdd,
    },
    {
      name: "Usuários",
      link: "/users",
      icon: FaUsers,
    },
  ];
}
```

### 8.1.1. Tela para Publicar Videoaulas

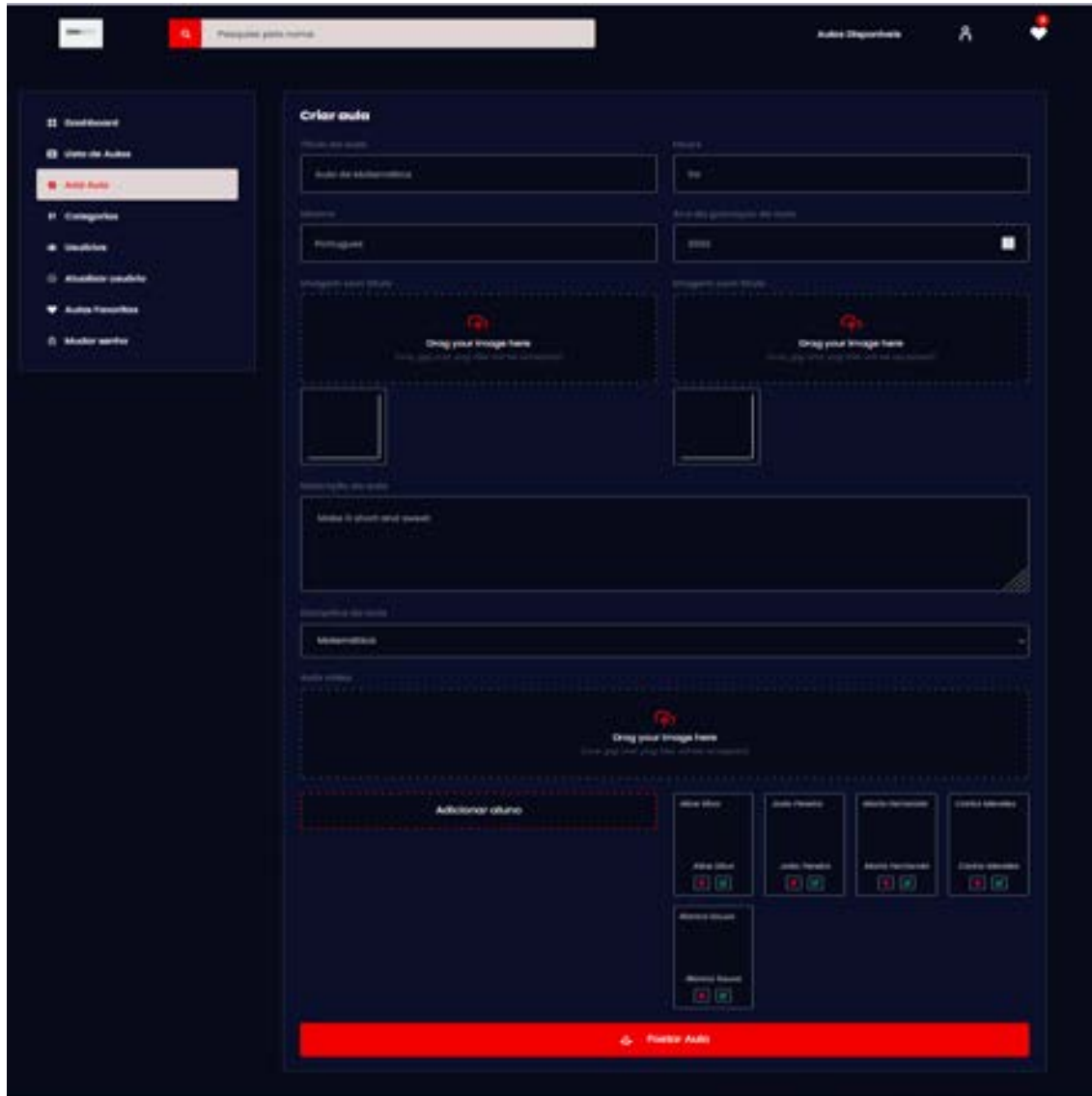
Na Figura 12, a tela para publicação de videoaulas inclui campos essenciais para a inserção de informações sobre o conteúdo. Os campos disponíveis são:

- **Título:** para identificar a videoaula.
- **Duração:** indica o tempo total da aula.
- **Data:** data de publicação da videoaula.
- **Arquivo de Imagem:** para o upload de uma imagem de capa representativa.
- **Descrição:** campo para descrever o conteúdo da videoaula.
- **Arquivo de Vídeo:** para upload do arquivo da videoaula.

- **Pessoas que Podem Acessar:** permite selecionar os usuários que terão acesso ao conteúdo.

Essa tela é importante para garantir que os educadores possam disponibilizar seus conteúdos de maneira organizada e eficiente, promovendo uma melhor experiência de aprendizado para os alunos.

Figura 12 – Tela de publicação de vídeos da plataforma de streaming.



Na Figura 13, o componente **AddAula** (linha 12) permite que usuários autorizados publiquem novas videoaulas na plataforma. Ao iniciar, ele configura estados para controlar a abertura do modal e os dados do elenco. O **useEffect** (linha 16) é utilizado para limpar os dados do elenco ao fechar o modal.

A interface inclui um formulário para inserir informações essenciais sobre a aula, como **Título**, **Duração**, **Idioma**, e **Ano da Gravação** (linhas 22-54). Os usuários também podem fazer upload de imagens e vídeos (linhas 56), com duas seções específicas para imagens: uma sem título e outra com título.

A seção de descrição (linha 87) permite adicionar um resumo da aula, enquanto a categoria da disciplina é selecionada através de um menu suspenso (linha 94). Para gerenciar o elenco, há um botão para adicionar alunos e uma lista de usuários disponíveis, onde cada usuário pode ser editado ou removido.

Por fim, o botão de envio finaliza a operação de postagem da aula(linha 144).

Figura 13: Part 1- Tela para publicação de videoaulas

```
1 import React, { useEffect, useState } from "react";
2 import Uploader from "../../Components/Uploader";
3 import { Input, Message, Select } from "../../Components/UsedInputs";
4 import SideBar from "../SideBar";
5 import { CategoriesData } from "../../Data/CategoriesData";
6 import { UsersData } from "../../Data/AulaData";
7 import { MdDelete } from "react-icons/md";
8 import { FaEdit } from "react-icons/fa";
9 import { ImUpload } from "react-icons/im";
10 import CastsModal from "../../Components/Modals/CastsModal";
11
12 function AddAula() {
13   const [modalOpen, setModalOpen] = useState(false);
14   const [cast, setCast] = useState(null);
15
16   useEffect(() => {
17     if (modalOpen === false) {
18       setCast();
19     }
20   }, [modalOpen]);
21
22   return (
23     <SideBar>
24       <CastsModal
25         modalOpen={modalOpen}
26         setModalOpen={setModalOpen}
27         cast={cast}
28       />
29     <div className="flex flex-col gap-6">
30       <h2 className="text-xl font-bold">Criar aula</h2>
31       <div className="w-full grid md:grid-cols-2 gap-6">
32         <Input
33           label="Título da aula"
34           placeholder="Aula de Matemática"
35           type="text"
36           bg={true}
37         />
38         <Input label="Horas" placeholder="1hr" type="text" bg={true} />
39       </div>
40
41       <div className="w-full grid md:grid-cols-2 gap-6">
42         <Input
43           label="Idioma"
44           placeholder="Português"
45           type="text"
46           bg={true}
47         />
48         <Input
49           label="Ano da gravação da aula"
50           placeholder="2022"
51           type="number"
52           bg={true}
53         />
54       </div>
55
56       </* IMAGENS */>
57       <div className="w-full grid md:grid-cols-2 gap-6">
58         </* Imagem sem título */>
59         <div className="flex flex-col gap-2">
60           <p className="text-border font-semibold text-sm">
61             Imagem sem título
62           </p>
63           <Uploader />
64         <div className="w-32 h-32 p-2 bg-main border border-border rounded">
```

**Figura 14- Part 2-** Tela para publicação de videoaulas, com campos para título, duração, idioma, ano de gravação, upload de imagem e vídeo, descrição, disciplina e gerenciamento do elenco de alunos.

```

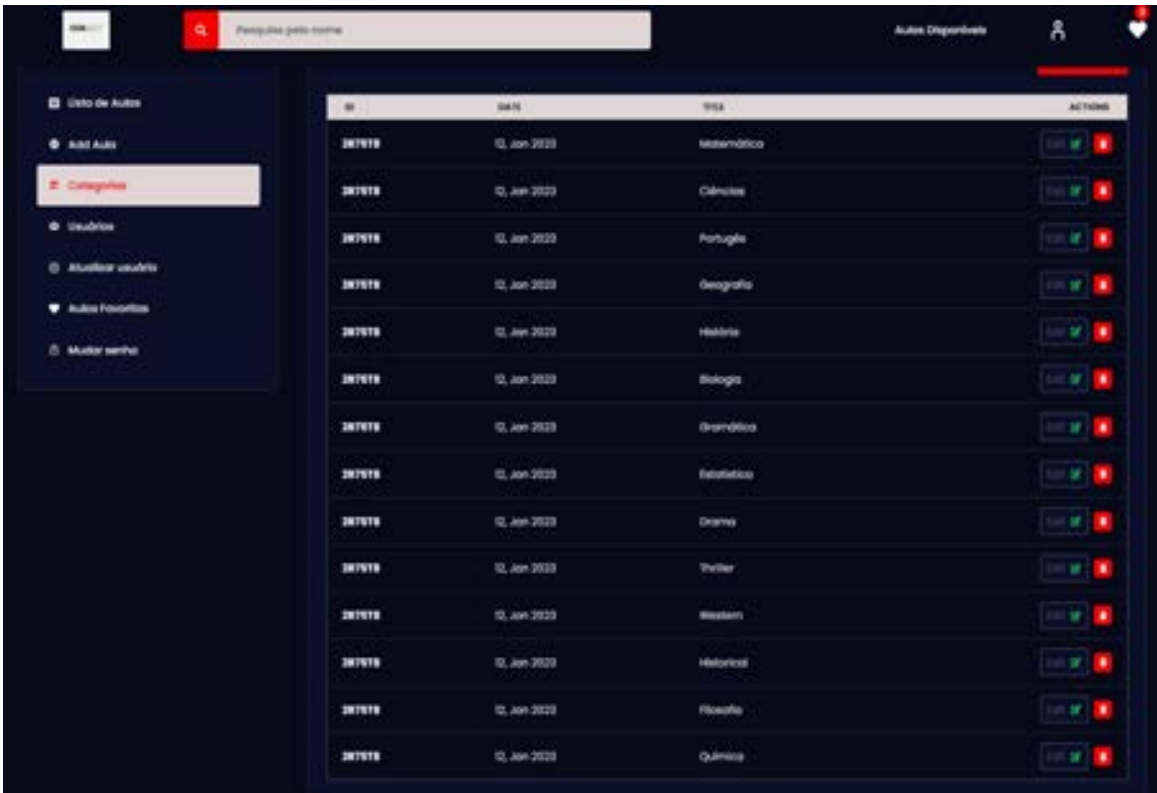
12 function AddAula() {
13
14   </div>
15 </div>
16 (/* DESCRIÇÃO */)
17 <message
18   label="Descrição da aula"
19   placeholder="Make it short and sweet"
20 />
21 (/* CATEGORIA */)
22 <div className="text-sm w-full">
23   <select label="Disciplina da aula" options={categoriasOuta} />
24 </div>
25 (/* VÍDEO DA AULA */)
26
27 <div className="flex flex-col gap-2 w-full">
28   <label className="text-border font-semibold text-sm">
29     Aula vídeo
30   </label>
31   <uploader />
32 </div>
33 (/* ALUNOS */)
34 <div className="w-full grid lg:grid-cols-2 gap-6 items-start">
35   <button
36     onClick={() => setModalOpen(true)}
37     className="w-full py-4 bg-main border border-submain border-dashed text-white rounded">
38     >
39     Adicionar aluno
40   </button>
41   <div className="grid 2xl:grid-cols-4 lg:grid-cols-3 sm:grid-cols-4 grid-cols-2 gap-4">
42     {UsersData.map((user, i) => {
43       <div
44         key={i}
45         className="p-2 italic text-xs text-text rounded flex-col bg-main border border-border">
46         >
47         <img
48           src={`/${images}/${user.image ? user.image : "user.png"}`}
49           alt={user.fullName}
50           className="w-full h-24 object-cover rounded mb-2"
51         />
52         <p>{user.fullName}</p>
53         <div className="flex-row mt-2 w-full gap-2">
54           <button className="w-6 h-6 flex-col bg-dry border border-border text-submain rounded">
55             <Delete />
56           </button>
57           <button
58             onClick={() => {
59               setCast(user);
60               setModalOpen(true);
61             }}
62             className="w-6 h-6 flex-col bg-dry border border-border text-green-600 rounded">
63             <Add />
64           </button>
65         </div>
66       </div>
67     })}
68   </div>
69 </div>
70 (/* SUBMIT */)
71 <button className="bg-submain w-full flex-row gap-6 font-medium text-white py-4 rounded">
72   <Upload /> Postar Aula
73 </button>
74 </div>

```

A tela de categorias, ilustrada na Figura 15, apresenta uma lista abrangente de todas as disciplinas disponíveis para a adição de videoaulas. Cada categoria é

exibida com informações relevantes, incluindo o ID da disciplina, a data de adição e opções para edição ou remoção. Essa interface permite que os administradores gerenciem facilmente as disciplinas, garantindo que as videoaulas sejam organizadas de acordo com as necessidades educacionais da plataforma.

*Figura 15 - Tela de categorias, exibindo disciplinas disponíveis para adição de videoaulas, com opções para editar ou remover matérias, além de mostrar o ID e a data de adição das disciplinas*



ID	DATA	MATÉRIA	AÇÕES
287618	10, Jan 2023	Matemática	[Edit] [Delete]
287618	10, Jan 2023	Ciências	[Edit] [Delete]
287618	10, Jan 2023	Português	[Edit] [Delete]
287618	10, Jan 2023	Geografia	[Edit] [Delete]
287618	10, Jan 2023	História	[Edit] [Delete]
287618	10, Jan 2023	Biologia	[Edit] [Delete]
287618	10, Jan 2023	Gramática	[Edit] [Delete]
287618	10, Jan 2023	Estadística	[Edit] [Delete]
287618	10, Jan 2023	Drama	[Edit] [Delete]
287618	10, Jan 2023	Thriller	[Edit] [Delete]
287618	10, Jan 2023	Western	[Edit] [Delete]
287618	10, Jan 2023	Historical	[Edit] [Delete]
287618	10, Jan 2023	Filosofia	[Edit] [Delete]
287618	10, Jan 2023	Química	[Edit] [Delete]

O código de categorias é mostrado na Figura 16, oferece uma interface para gerenciar as disciplinas disponíveis na plataforma. O componente principal é a função `Categories`, que utiliza a biblioteca `React` para renderizar a interface. O estado do componente é gerenciado com os hooks `useState` e `useEffect`. A variável `modalOpen` controla a visibilidade do modal para adicionar ou editar categorias (linha 10). Quando o botão "Create" é clicado (linha 42), a função `setModalOpen(true)` é chamada, abrindo o modal para permitir que os usuários criem uma nova categoria.



O componente Table2 (linha 47) exibe a lista de categorias disponíveis, recebendo dados da constante CategoriesData. A função OnEditFunction é chamada quando o usuário deseja editar uma categoria, passando o ID da categoria para o estado correspondente e abrindo o modal de edição.

**Figura 16** - Código de categorias que apresenta todas as disciplinas disponíveis para adição de videoaulas, com opções para criar novas categorias ou editar as existentes.

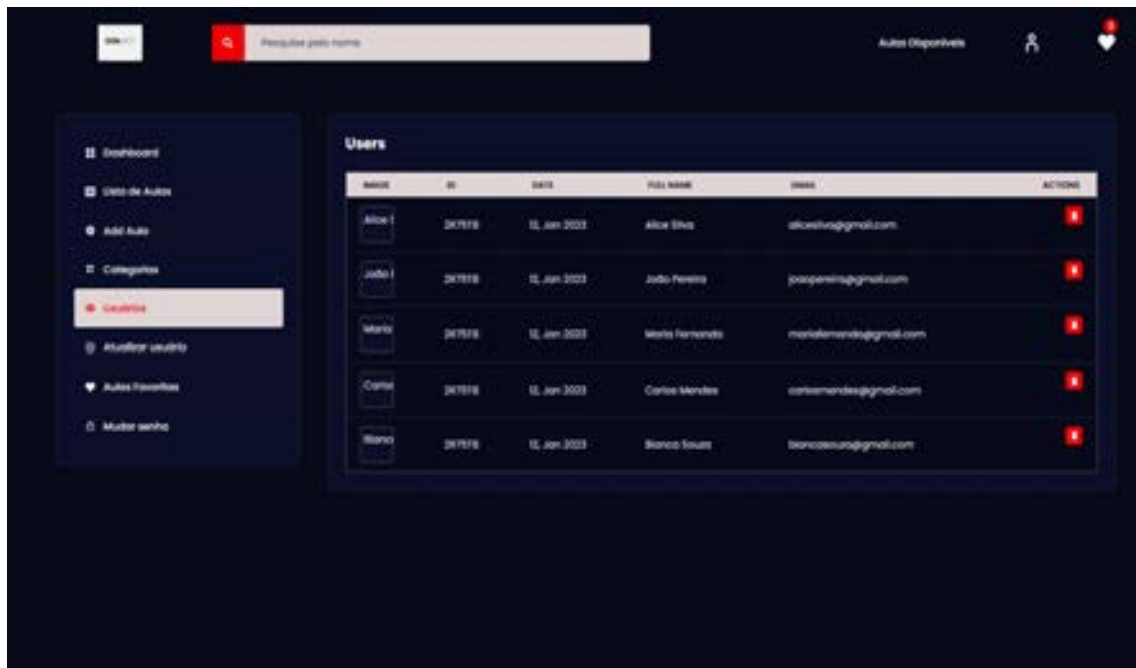
```
src / Screens / Dashboard / Admin / Categories.js / Categories
1 import React, { useEffect, useState } from "react";
2 import { HiPlusCircle } from "react-icons/hi";
3 import Table2 from "../../Components/Table2";
4 import Sidebar from "../Sidebar";
5 import { CategoriesData } from "../../Data/CategoriesData";
6 import CategoryModal from "../../Components/Modals/CategoryModal";
7
8 function Categories() {
9   // Define o estado para controlar a abertura do modal e a categoria selecionada para edição
10  const [modalOpen, setModalOpen] = useState(false);
11  const [category, setCategory] = useState();
12
13  // Função para abrir o modal e selecionar uma categoria para edição
14  const OnEditFunction = (id) => {
15    setCategory(id);
16    setModalOpen(!modalOpen);
17  };
18
19  // Hook useEffect que reseta a categoria selecionada quando o modal é fechado
20  useEffect(() => {
21    if (modalOpen === false) {
22      setCategory();
23    }
24  }, [modalOpen]);
25
26  return (
27    <Sidebar>
28      /* Modal para criação ou edição de categoria */
29      <CategoryModal
30        modalOpen={modalOpen}
31        setModalOpen={setModalOpen}
32        category={category}
33      />
34      <div className="flex flex-col gap-6">
35        /* Título e botão para abrir o modal de criação de nova categoria */
36        <div className="flex btn gap-2">
37          <h2 className="text-xl font-bold">Categories</h2>
38          <button
39            onClick={() => setModalOpen(true)}
40            className="bg-subMain flex-rows gap-4 font-medium transitions hover:bg-main border
41          >
42            <HiPlusCircle /> Create
43          </button>
44        </div>
45
46        /* Componente de tabela que exibe as categorias e possibilita a edição */
47        <Table2
48          data={CategoriesData} // Dados das categorias para preencher a tabela
49          users={false} // Indica que não são dados de usuários
50          OnEditFunction={OnEditFunction} // Função que habilita a edição ao clicar em um item
51        />
52      </div>
53    </Sidebar>
54  );

```



Na Figura 17 é exibida a tela de usuário contendo todos os usuários registrados na plataforma, e com opção de gerenciamento de cada um, para facilitar quem pode ver o conteúdo.

*Figura 17 - Tela de usuários, exibindo a lista de pessoas cadastradas na plataforma de streaming*



A tela de usuários, representada na Figura 18, exibe a lista de usuários cadastrados no sistema. O componente Users, que inicia na linha 6 e termina na linha 16, utiliza o SideBar para a navegação.

Na linha 11, o título "Usuários" é exibido, e na linha 14, o componente Table2 apresenta os dados dos usuários da variável UserData, configurada com users={true} para formatação específica.

Figura 18 - Tela de usuários mostrando a lista de usuários cadastrados.

```
my-app > src > Screens > Dashboard > Admin > JS Users.js > [🔗] default
1  import React from "react";
2  import Table2 from "../../Components/Table2";
3  import SideBar from "../SideBar";
4  import { UsersData } from "../../Data/AulaData";
5
6  function Users() {
7    return (
8      <SideBar>
9        <div className="flex flex-col gap-6">
10         /* Título da seção de usuários */
11         <h2 className="text-xl font-bold">Usuários</h2>
12
13         /* Tabela que exibe a lista de usuários */
14         <Table2 data={UsersData} users={true} />
15       </div>
16     </SideBar>
17   );
18 }
19
20 export default Users;
```

A Figura 19 ilustra a tela de aulas favoritas apresenta uma lista das aulas que o usuário selecionou como favoritas. Nela, os usuários podem gerenciar suas aulas, como removê-las da lista de favoritas ou realizar o download para assistir offline. A interface é projetada para ser intuitiva, permitindo um fácil acesso e controle sobre as aulas preferidas.

Figura 19 - Tela de aulas favoritas, exibindo todas as aulas adicionadas aos favoritos



Na Figura 20, vemos o componente FavoritesAulas exibe a tela de aulas favoritas. Ele inclui:

- Importações (Linhas 1-5): Carrega o React, o componente Table e a barra lateral (SideBar).
- Componente (Linha 7): Define o componente funcional.
- Layout (Linhas 9-22): Mostra o título "Aulas Favoritas" e um botão "Deletar Tudo".
- Tabela (Linha 20): Renderiza a lista de aulas favoritas usando o componente Table.

Figura 20 - Tela de Aulas Favoritas, permitindo gerenciar e deletar aulas favoritas.

```
my-app > src > Screens > Dashboard > FavoritesMovies > FavoritesAulas
1 // Importa React, o componente de tabela, a barra lateral e dados das aulas
2 import React from "react";
3 import Table from "../../Components/Table";
4 import SideBar from "../SideBar";
5 import { Aulas } from "../../Data/AulaData";
6
7 function FavoritesAulas() {
8   return (
9     <SideBar>
10      <div className="flex flex-col gap-6">
11        <div className="flex btn gap-2">
12          /* Título da seção de aulas favoritas */
13          <h2 className="text-xl font-bold">Aulas Favoritas</h2>
14          <button className="bg-main font-medium transitions hover:bg-subMain border border-subMain">
15            Deletar Tudo
16          </button>
17        </div>
18
19        /* Tabela que exibe a lista de aulas favoritas */
20        <Table data={Aulas} admin={false} />
21      </div>
22    </SideBar>
23  );
24 }
25
26 export default FavoritesAulas;
```

A Figura 21 exibe a tela de descrição da vídeo aula, contendo o nome da aula, disciplina e uma breve descrição, também com opção de assistir online ou por download

Figura 21 - Tela de Descrição da Vídeo Aula, com detalhes sobre a aula e opções de acesso.



A Figura 22 ilustra o código do componente da tela SingleAula exibe os detalhes e descrição de uma vídeo aula, incluindo um modal para compartilhamento (linha 14), informações da aula (linha 25) e avaliações (linha 31). Também apresenta sugestões de aulas relacionadas na mesma categoria (linha 34).

Figura 22 - Código da Tela de Descrição da Vídeo Aula, com detalhes e sugestões de aulas relacionadas.

```
my-app > src > Screens > SingleMovie.js > default
1: // Importa React, hooks de estado e parâmetros de rota, componentes de detalhes, slider, avaliações, títulos, dados de aulas e layout
2: import React, { useState } from "react";
3: import { useParams } from "react-router-dom";
4: import AulaCards from "../Components/Single/AulaCards";
5: import AulaInfo from "../Components/Single/AulaInfo";
6: import AulaRates from "../Components/Single/AulaRates";
7: import Titles from "../Components/Titles";
8: import { Aulas } from "../Data/AulaData";
9: import Layout from "../Layout/Layout";
10: import { BsCollectionFill } from "react-icons/bs";
11: import Aula from "../Components/Aula";
12: import ShareAulaModal from "../Components/Modals/ShareModal";
13:
14: function SingleAula() {
15:   const [modalOpen, setModalOpen] = useState(false);
16:   const { id } = useParams();
17:   const aula = Aulas.find(aula => aula.name === id);
18:   const RelatedAulas = Aulas.filter(a => a.category === aula.category);
19:
20:   return (
21:     <Layout>
22:       <ShareAulaModal
23:         modalOpen={modalOpen}
24:         setModalOpen={setModalOpen}
25:         aula={aula}
26:       />
27:       <AulaInfo aula={aula} setModalOpen={setModalOpen} />
28:       <div className="container mx-auto min-h-screen px-2 my-6">
29:         <AulaCards />
30:         < /* Avaliações da aula */
31:         <AulaRates aula={aula} />
32:         < /* Aulas relacionadas */
33:         <div className="my-16">
34:           <Titles title="Aulas relacionadas" icon={BsCollectionFill} />
35:           <div className="grid sm:mt-10 mt-6 xl:grid-cols-4 2xl:grid-cols-5 lg:grid-cols-3 sm:grid-cols-2 gap-6">
36:             <RelatedAulas.map(aula, index) => (
37:               <Aula key={index} aula={aula} />
38:             )
39:           </div>
40:         </div>
41:       </div>
42:     </Layout>
43:   );
44: }
45:
46: export default SingleAula;
```

A Figura 23 ilustra a tela de visualização online da plataforma, onde abre um player e tem a possibilidade de assistir a vídeo aula, ou também se preferir, contém a opção de download e favoritar aula.

Figura 23 - Tela de Visualização Online da Vídeo Aula, com player integrado e opções de download e favoritar.



Nas Figuras 24 e 25 ilustra o código do componente WatchPage que renderiza a tela de visualização da vídeo aula, com opções para assistir online, favoritar e baixar. O código utiliza useParams (linha 9) para capturar o identificador da aula e encontrar o vídeo correspondente no array Aulas.

Ao carregar a tela, o usuário vê o título da aula e os botões de ações (linha 23). O botão de favoritos (FaHeart) e o botão de download (FaCloudDownloadAlt) estão dentro de uma área de ação flexível (linha 27). Quando o usuário clica no botão de play (FaPlay), a função setPlay é chamada, trocando a visualização para o player de vídeo com autoplay ativado.

Figura 24 – Código da tela de Visualização de Aula, contendo player de vídeo, botões de download e de favoritos.

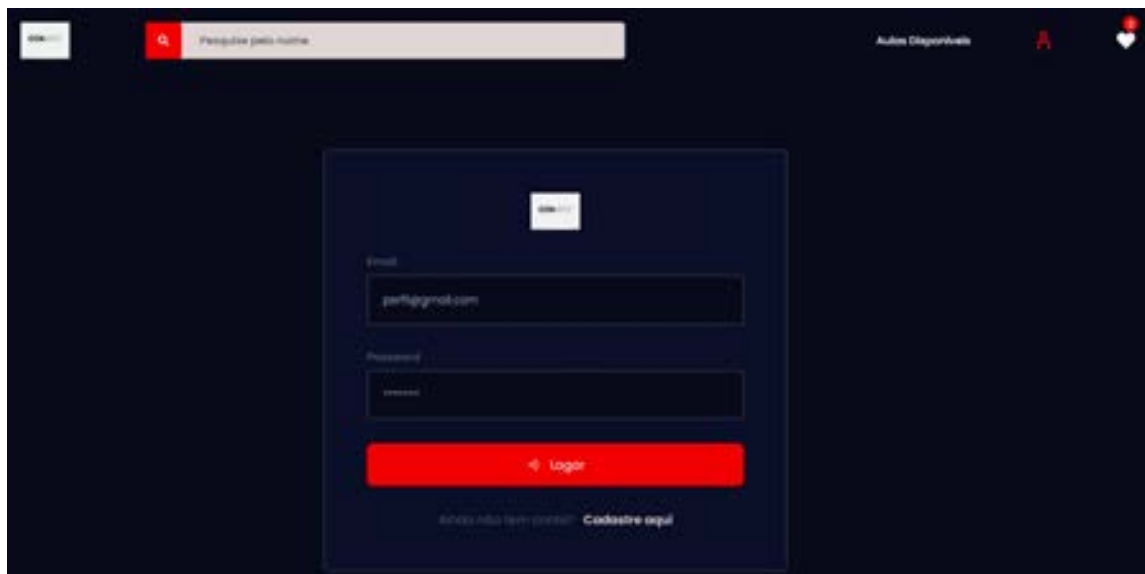
```
src > Screens > WatchPage.js > WatchPage
1  import React, { useState } from "react";
2  import { useParams, Link } from "react-router-dom";
3  import { Aulas } from "../Data/AulaData";
4  import Layout from "../Layout/Layout";
5  import { SiArrowBack } from "react-icons/si";
6  import { FaCloudDownloadAlt, FaHeart, FaPlay } from "react-icons/fa";
7
8  function WatchPage() {
9    let { id } = useParams();
10   const aula = Aulas.find(aula => aula.name === id);
11   const [play, setPlay] = useState(false);
12   return (
13     <Layout>
14       <div className="container mx-auto lg:px-6 md:px-12">
15         <div className="flex-bits flex-wrap md-6 gap-2 lg-main rounded border border-gray-500 p-6">
16           <Link
17             to={`/aula/${aula?.name}`}
18             className="md:text-xl text-sm flex gap-3 items-center font-bold text-gray-900">
19             <SiArrowBack /> {aula?.name}
20           </Link>
21         </div>
22         <div className="flex-bits sm:px-auto w-full gap-5">
23           <button className="bg-white hover:text-submain transition bg-opacity-30 text-white rounded px-4 py-3 text-sm">
24             <FaHeart />
25           </button>
26           <button className="bg-submain flex-row gap-2 hover:text-main transition text-white rounded px-8 font-medium py-3 text-sm">
27             <FaCloudDownloadAlt /> Download Vídeo Aula
28           </button>
29         </div>
30       </div>
31     </Layout>
32     < /* assistir aulas : */
33     <play ? <
34       <video controls autoPlay={play} className="w-full h-full rounded">
35         <source
36           src="/images/aula.mp4"
37           type="video/mp4"
38           title={aula?.name}
39         />
40       </video>
41     </play> : <
```

**Figura 25** – Código da tela de Visualização de Aula, contendo player de vídeo, botões de download e de favoritos.

```
32      {/* assistir aula s */}
33      {play ? {
34        <video controls autoPlay={play} className="w-full h-full rounded">
35          <source
36            src="/images/aula.mp4"
37            type="video/mp4"
38            title={aula?.name}
39          />
40        </video>
41      } {
42        <div className="w-full h-screen rounded-lg overflow-hidden relative">
43          <div className="absolute top-0 left-0 bottom-0 right-0 bg-main bg-opacity-30 flex-clip">
44            <button
45              onClick={() => setPlay(!play)}
46              className="bg-white text-subMain flex-clip border border-subMain rounded-full w-20 h-20 font-medium text-xl"
47            >
48              <FaPlay />
49            </button>
50          </div>
51          <img
52            src={
53              aula?.image
54                ? "/images/aulas/${aula?.image}"
55                : "images/cover.png"
56            }
57            alt={aula?.name}
58            className="w-full h-full object-cover rounded-lg"
59          />
60        </div>
61      } }
62    </div>
63  </Layout>
64 }
65 }
66 }
67 export default WatchPage;
```

A Figura 26 exibe a tela de login, que permite que o usuário insira seu e-mail e senha registrados para acessar a plataforma. O formulário contém campos de entrada para e-mail e senha, e um botão para enviar as credenciais. Em caso de falha no login, são exibidas mensagens de erro orientando o usuário a verificar suas informações.

**Figura 26** - Tela de Login, com campos para e-mail e senha e botão de envio para acessar a plataforma.



A Figura 27 ilustra o código da tela de Login permite ao usuário inserir seu e-mail e senha para acessar a plataforma. A logo é exibida no topo, seguida pelos campos de entrada para e-mail e senha, criados com o componente `<Input>` que está dentro do Layout (linha 9 – 43). Um botão leva o usuário ao dashboard após o login, e um link adicional oferece a opção de cadastro para novos usuários (linha 37).

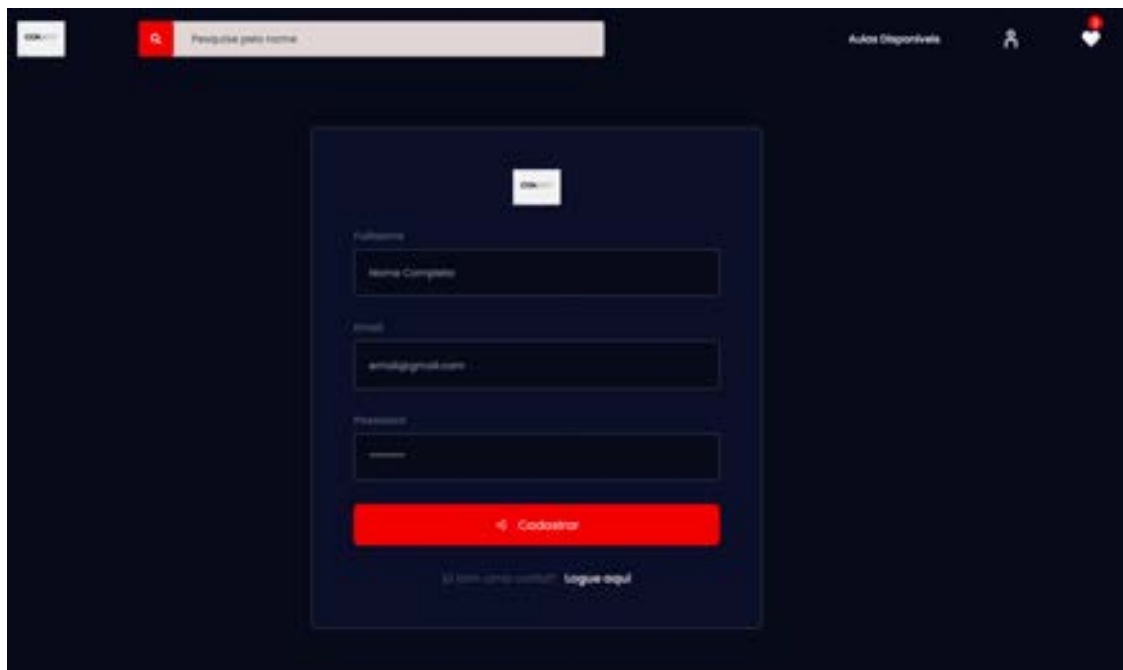
**Figura 27** – Código da tela de Login da plataforma.



```
src > Screens > Login.js > ...
1 import React from "react";
2 import { Link } from "react-router-dom";
3 import { Input } from "../Components/UsedInputs";
4 import Layout from "../Layout/Layout";
5 import { FiLogin } from "react-icons/fi";
6
7 function login() {
8   return (
9     <Layout>
10       <div className="container mx-auto px-2 py-24 flex-colo">
11         <div className="w-full 2xliw-2/5 gap-8 flex-colo p-8 sm:p-14 md:w-3/5 bg-dry rounded-lg border border-border">
12           
17           <Input
18             label="Email"
19             placeholder="perfil@gmail.com"
20             type="email"
21             bg={true}
22           />
23           <Input
24             label="Password"
25             placeholder="*****"
26             type="password"
27             bg={true}
28           />
29           <Link
30             to="/dashboard"
31             className="bg-subMain transitions hover:bg-main flex-row gap-4 text-white p-4 rounded-lg w-full"
32           >
33             <FiLogin /> <span>Login</span>
34           </Link>
35           <div className="text-center text-border">
36             Ainda não tem conta? (" ")
37             <Link to="/register" className="text-dryGray font-sans font-weight-normal ml-2">
38               Cadastre aqui
39             </Link>
40           </div>
41         </div>
42       </div>
43     </Layout>
44   );
45 }
46
47 export default login;
48
```

A tela de registro é exibida na Figura 28, para ter a possibilidade de fazer o login e acessar a plataforma e seu menu administrativo(dashboard)

Figura 28 – Tela de Registro da plataforma de streaming



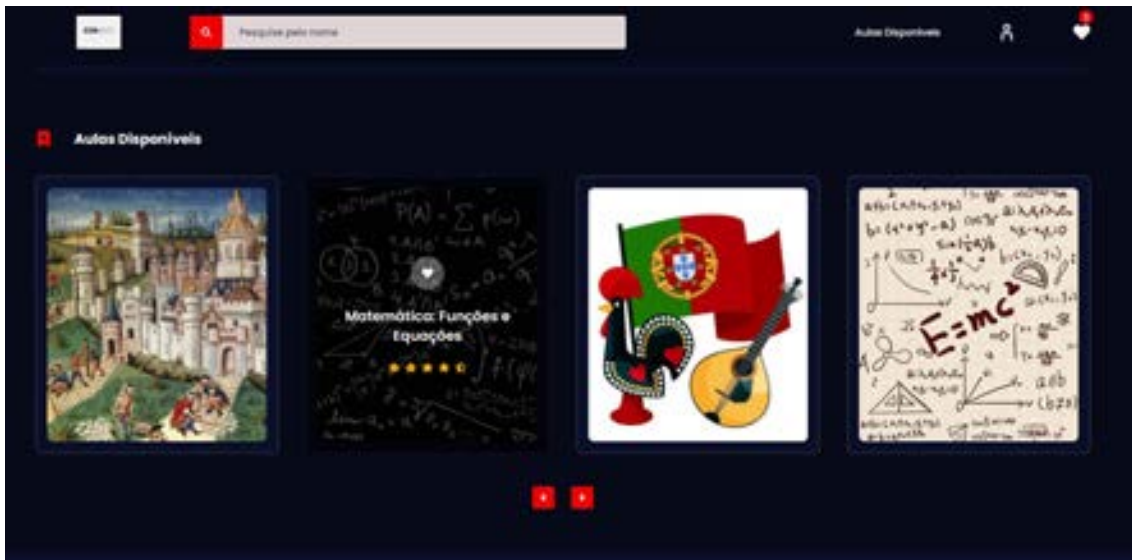
Na Figura 29 vemos o código da tela de registro permite que novos usuários se cadastrem na plataforma para, posteriormente, fazerem login e acessarem o dashboard administrativo. O código começa importando componentes essenciais, como o layout principal e campos de entrada personalizados (Input) nas linhas iniciais. Na linha 7, é definida a função Register, que retorna a estrutura HTML da página. No formulário, cada Input recebe um label e um placeholder para orientar o usuário no preenchimento de seu nome completo, e-mail e senha (linhas 16 a 36). Ao final do formulário, um botão de cadastro redireciona o usuário para o dashboard (linha 36), enquanto um link opcional leva o usuário para a página de login caso ele já possua uma conta (linha 42).

**Figura 29 – Código da tela de Registro da plataforma de streaming**

```
src > Screens > Register.js > default
1  import React from "react";
2  import { Link } from "react-router-dom";
3  import { Input } from "../Components/UsedInputs";
4  import Layout from "../Layout/Layout";
5  import { FiLogin } from "react-icons/fi";
6
7  function Register() {
8    return (
9      <Layout>
10     <div className="container mx-auto px-2 my-24 flex-colo">
11       <div className="w-full 2xl:w-2/5 gap-8 flex-colo p-8 sm:p-14 md:w-3/5 bg-dry rounded-lg border">
12         
17         <Input
18           label="FullName"
19           placeholder="Nome Completo"
20           type="text"
21           bg={true}
22         />
23         <Input
24           label="Email"
25           placeholder="email@gmail.com"
26           type="email"
27           bg={true}
28         />
29         <Input
30           label="Password"
31           placeholder="*****"
32           type="password"
33           bg={true}
34         />
35         <Link
36           to="/dashboard"
37           className="bg-subMain transitions hover:bg-main flex-row gap-4 text-white p-4 rounded"
38         >
39           <FiLogin /> Cadastrar
40         </Link>
41         <p className="text-center text-border">
42           Je tem uma conta{" "}
43           <Link to="/login" className="text-dryGray font-semibold ml-2">
44             logue aqui
45           </Link>
46         </p>
47       </div>
48     </div>
49   </Layout>
50 );
51 }
52
53 export default Register;
54
```

A Figura 30 exibe a tela de vídeo aulas disponíveis apresenta ao usuário uma lista completa de aulas atualmente acessíveis para visualização online na plataforma, organizadas para facilitar a navegação e a busca por conteúdo.

Figura 30 - Tela de vídeo aulas disponíveis, onde o usuário pode visualizar todas as aulas acessíveis na plataforma para assistir online.



A Figura 30 exibe o código do componente ClassesPage é responsável por exibir a lista de aulas disponíveis, com um botão para carregar mais itens progressivamente. Utiliza o componente Layout (linha16) para a estrutura básica da página e Filters (linha 18) para filtros de busca. As aulas são renderizadas em um grid, onde o número inicial de aulas exibidas é controlado pela constante maxPage. A função HandleLoadingMore (linha31) permite que o usuário carregue mais aulas ao clicar no botão "Carregar Mais". O ícone de CgSpinner (linha 34) exibe uma animação de carregamento enquanto o botão é acionado.

Figura 31 – Código da tela de vídeo aulas disponíveis

```

1 import React, { useState } from "react";
2 import Filters from ".../components/filters";
3 import Layout from ".../components/layout";
4 import Class from ".../components/class";
5 import { Classes } from ".../data/classes";
6 import { CgSpinner } from "react-icons/cg";
7
8 function ClassesPage() {
9   const [maxPage] = useState(12);
10   const [page, setPage] = useState(1);
11   const [isLoadingMore] = useState(false);
12
13   const handleLoadMore = () => {
14     setPage(page + maxPage);
15   };
16
17   return (
18     <Layout>
19       <div className="min-height-screen container mx-auto px-2 py-4">
20         <Filters />
21         <div className="text-center font-medium my-4">
22           Total <span className="font-bold">{Classes.length}</span> <span>{"/page"}</span>
23           <span>{page}</span>
24         </div>
25         <div className="grid sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4">
26           {Classes.slice((page-1)*maxPage, page*maxPage).map((cls, index) => {
27             <Class key={index} aula={cls} />
28           })}
29         </div>
30         <div>
31           <span>{isLoadingMore}</span>
32           <button
33             onClick={handleLoadMore}
34             className="flex items gap-1 text-white px-2 py-2 rounded font-medium border-2 border-solid">
35             Carregar Mais <CgSpinner className="animate-spin" />
36           </button>
37         </div>
38       </div>
39     </Layout>
40   );
41 }

```

## **8. Ferramentas e Métodos**

As tecnologias utilizadas no desenvolvimento do projeto incluem: React para a construção da interface, Tailwind CSS para estilização. A escolha dessas ferramentas se deu pela sua eficiência, robustez e forte apoio da comunidade.

## **9. Desenvolvimento de sistema**

O desenvolvimento do sistema foi conduzido de forma iterativa e incremental, utilizando a metodologia ágil. Esse processo foi estruturado em fases, com entregas contínuas e revisões frequentes para garantir que os requisitos evoluíssem de acordo com as necessidades dos usuários e stakeholders. A abordagem ágil permitiu maior flexibilidade, promovendo a adaptação às mudanças e facilitando a colaboração entre as equipes de desenvolvimento, design e testes.

A arquitetura do sistema foi dividida em duas camadas principais: frontend e backend. O frontend foi desenvolvido utilizando o React, uma biblioteca JavaScript que permite a construção de interfaces de usuário dinâmicas e reativas. O React foi escolhido por sua flexibilidade e eficiência na gestão do estado da aplicação, permitindo o desenvolvimento de uma interface interativa e de fácil manutenção. O uso de componentes reutilizáveis no React facilitou a escalabilidade da aplicação, enquanto o roteamento dinâmico e a renderização no lado do cliente melhoraram a performance geral.

## **10. Resultados e Discussão**

O protótipo desenvolvido cumpriu com êxito os objetivos estabelecidos, incorporando as principais funcionalidades previstas no planejamento. A plataforma mostrou-se altamente acessível, oferecendo uma experiência consistente em diversos dispositivos e navegadores. Além disso, foram realizadas análises de desempenho, que confirmaram a eficiência do sistema em termos de tempo de resposta e estabilidade, assegurando sua robustez e capacidade de atender às demandas dos usuários.

## 11. Considerações Finais

Este trabalho teve como objetivo o desenvolvimento de um protótipo funcional da plataforma de streaming de videoaulas denominada **EDUSTREAM**, projetada para atender às necessidades de professores e alunos. A plataforma permite o upload e a visualização de conteúdos educacionais em formato de vídeo, facilitando o acesso à educação de qualidade, especialmente em um contexto de ensino a distância.

O desenvolvimento do sistema foi orientado por técnicas, metodologias e linguagens de modelagem de engenharia de software, incluindo TAP (Termo de Abertura de Projeto), EAP (Estrutura Analítica do Projeto), SWOT (Strengths, Weaknesses, Opportunities e Threats), BPMN (Business Process Model and Notation) e UML (Unified Modeling Language). A escolha da biblioteca ReactJS e da linguagem de folha de estilo CSS possibilitou a criação de uma interface responsiva e amigável, que atende tanto a dispositivos móveis quanto a desktops.

O projeto busca atender à crescente demanda por plataformas educacionais online, ampliando a disponibilidade para aprender. Olhando para o futuro, planeja-se a inclusão de funcionalidades adicionais, como implementação do back-end, quizzes interativos, integração com Banco de dados, autenticação de usuário e fóruns de discussão, que poderão enriquecer a experiência de aprendizado e fomentar uma maior interação entre os usuários.

## Referências

- Booch, G., Rumbaugh, J., & Jacobson, I. (2005).** *The Unified Modeling Language User Guide*. Addison-Wesley.
- Brown, T. (2009).** *Change by Design: How Design Thinking Creates New Alternatives for Business and Society*. Harper Business.
- Chiavenato, I. (2014).** *Teoria Geral da Administração: uma Abordagem Abrangente*. Elsevier.
- Clements, P., & Kazman, R. (2003).** *Software Architecture in Practice*. Addison-Wesley.
- Jacobson, I. (1992).** *Object-Oriented Software Engineering: a Use Case Driven Approach*. Addison-Wesley.
- Kotonya, G., & Sommerville, I. (1998).** *Requirements Engineering: Processes and Techniques*. Wiley.
- Larman, C. (2004).** *Applying UML and Patterns: an Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall.
- Meta. (2023).** *React Documentation*. Disponível em: <https://react.dev>.
- OMG (Object Management Group). (2011).** *BPMN 2.0 Handbook*.
- PMI (Project Management Institute). (2017).** *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. PMI.
- Pressman, R. S. (2011).** *Software Engineering: a Practitioner's Approach*. McGraw-Hill.
- Ribeiro, A., Santos, L., & Almeida, M. (2021).** *Educação a Distância no Brasil: Desafios e Perspectivas*. Revista Brasileira de Educação Online, 17(2), 45-62.
- Rosenfeld, L., Morville, P., & Arango, J. (2015).** *Information Architecture: for the Web and Beyond*. O'Reilly Media.
- Schmidt, C. (2008).** *Prototyping for Designers: developing the New Standard for Design Practice*. Design Books.
- Silva, G. H. A. (2024).** *Diagrama de Caso de Uso: aplicações em Sistemas Educacionais*. Revista de Modelagem de Software, 15(3), 45-57.
- Snyder, C. (2003).** *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann.
- Sommerville, I. (2011).** *Software Engineering*. Pearson.

**Vargas, R.** (2014). *Manual Prático do Plano de Projeto: Como Preparar um Plano de Projeto Passo a Passo*. Brasport.

**Vercel.** (2023). *Next.js Documentation*. Disponível em: <https://nextjs.org/docs>.

## Apêndice A

### Perguntas e Respostas da Entrevista

#### **1. Qual a funcionalidade mais importante em uma plataforma de videoaulas para você?**

**Professor:** "A opção de upload de videoaulas e de organizar o conteúdo por categorias ou disciplinas."

#### **2. Como você prefere organizar ou visualizar o conteúdo na plataforma?**

**Aluno:** "Eu prefiro uma lista organizada por disciplinas, para encontrar as videoaulas específicas de cada matéria."

#### **3. Você considera importante a opção de fazer comentários nas videoaulas? Por quê?**

**Aluno:** "Sim, porque assim posso tirar dúvidas ou compartilhar ideias com os outros alunos e com o professor."

#### **4. Você considera importante ter uma área para avaliações de videoaulas? Por quê?**

**Professor:** "Acho útil, pois isso me permite saber quais aulas estão ajudando mais os alunos."

#### **5. Como você prefere a visualização da plataforma?**

**Aluno:** "Para mim, uma interface simples e que funcione bem no celular é essencial, já que assisto bastante pelo celular."

#### **6. Você considera importante ter a opção de notificações para novos conteúdos?**

**Professor:** "Sim, para avisar os alunos sobre novos vídeos ou atualizações importantes."

#### **7. Gostaria de ver recursos adicionais como quizzes ou materiais complementares junto aos vídeos?**

**Aluno:** "Sim, os quizzes ajudariam a testar o que aprendi."



**8. Como você vê a importância da segurança e privacidade na plataforma?**

**Aluno:** "É importante que meus dados fiquem protegidos e que só eu tenha acesso ao meu histórico de aulas."

**9. Você teria sugestões para melhorar a experiência na plataforma?**

**Aluno:** "Poderia haver uma função de download para assistir offline."