MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE RONDÔNIA
CAMPUS VILHENA
PÓS-GRADUAÇÃO LATO SENSU EM DESENVOLVIMENTO WEB

DESENVOLVIMENTO DE UMA PLATAFORMA DE CURSOS MOOC

ERICKY MORENO MAMEDE
FABRÍCIO GABRIEL MIRANDA DA SILVA
LUCAS MATOS COELHO
JULIANO DE MELLO

VILHENA
2025

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE RONDÔNIA
CAMPUS VILHENA

DESENVOLVIMENTO DE UMA PLATAFORMA DE CURSOS MOOC

ERICKY MORENO MAMEDE
FABRÍCIO GABRIEL MIRANDA DA SILVA
LUCAS MATOS COELHO
JULIANO DE MELLO

Produto apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia, como requisito avaliativo para conclusão do curso de Pós-graduação em Desenvolvimento WEB, sob a orientação do Professor Wesley Jhones Ramos Rolim.

VILHENA

2025

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO

Desenvolvimento de uma plataforma de cursos MOOC / Ericky Moreno Mamede... [et al] / Vilhena, 2025.
26f. : il.

Orientador(a):Prof. Me.Wesley Jhones Ramos Rolim
Trabalho de Conclusão de Curso (Pós-Graduação Lato Sensu em Desenvolvimento Web) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Vilhena, 2025.

1.Silva, Fabrício Gabriel Miranda da 2. Coelho, Lucas Matos 3. Mello, Juliano de I. Rolim, Wesley Jhones Ramos (orient.) II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Rosilene Maria do Couto Marques, CRB-11/321

1. INTRODUÇÃO

A trajetória histórica do ensino a distância (EAD) remonta a eventos significativos que moldaram sua evolução ao longo do tempo. Na Suécia, em 1833, a Universidade da cidade de Lund inovou ao introduzir um curso de composição por correspondência, marcando um notável ponto de partida. Pouco depois, em 1840, na Inglaterra, uma iniciativa semelhante floresceu com um curso de Taquigrafia dedicado à transcrição de passagens bíblicas. Não apenas instrutivo, esse curso conduzido pelo professor Isaac Pitman também encorajava os alunos a trocarem postais contendo textos abreviados, consolidando assim uma prática pedagógica precursora do aprendizado remoto. Esses eventos históricos não apenas ilustram a origem do EAD, mas também estabelecem um contexto enriquecedor para compreender a trajetória subsequente desse método educacional.

E hoje, com a revolução da internet e da tecnologia, as barreiras geográficas que antes limitavam o acesso ao conhecimento foram derrubadas, possibilitando que o ensino a distância (EAD) alcance todas as partes do mundo. Essa modalidade educacional tem se consolidado como uma tendência em constante crescimento na capacitação das pessoas. O Censo da Educação Superior, divulgado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep) e pelo Ministério da Educação (MEC), Entre 2011 e 2021, o número de ingressantes em cursos superiores de graduação, na modalidade de educação a distância (EaD), aumentou 474%. No mesmo período, a quantidade de ingressantes em cursos presenciais diminuiu 23,4%. (https://corta.link/KyG3n)

Os benefícios do EAD são notáveis, oferecendo aos alunos a flexibilidade de ajustar seu ritmo de aprendizado de acordo com seus horários e a conveniência de estudar em qualquer local com acesso à internet. Essa flexibilidade torna o EAD uma escolha atraente para estudantes de todos os níveis de ensino.

Além disso, o ensino digital não está limitado ao ensino superior, mas se estende aos cursos de formação complementar. Esses cursos se destacam por sua natureza objetiva e curta duração, proporcionando uma maneira eficaz de adquirir novas habilidades para aprimorar a formação profissional.

Os cursos complementares são oferecidos tanto por instituições de ensino superior quanto por empresas privadas especializadas na área. As empresas muitas vezes cobram por assinaturas, por curso ou simplesmente pelo certificado de conclusão. Por outro lado, as instituições de ensino tendem a disponibilizar cursos e certificados gratuitos.

Um exemplo inspirador desse comprometimento com a educação é o Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), que, por meio do Laboratório de Fábrica de Software (FSLab), está empenhado em criar uma nova plataforma de cursos gratuitos para apoiar a formação de seus alunos. Inicialmente, a plataforma estará disponível apenas internamente no instituto, mas existe a perspectiva de abri-la à comunidade no futuro.

Com essa iniciativa, busca-se atender de maneira mais eficaz às necessidades dos alunos e profissionais do instituto, incluindo a exigência das horas complementares necessárias para a conclusão dos cursos superiores. O desafio que se apresenta é como melhorar o acesso dos alunos a cursos complementares gratuitos de qualidade e com certificação reconhecida, garantindo que a educação a distância continue a desempenhar um papel vital no desenvolvimento das pessoas, independentemente de sua localização ou nível de ensino.

2. PROCESSOS E MÉTRICAS

Este estudo foi conduzido no âmbito do programa de pós-graduação em Desenvolvimento Web, tendo como propósito a concepção de uma plataforma de cursos online denominada "FS MOOC". O processo de desenvolvimento do referido sistema estendeu-se por vários meses e foi realizado por meio de encontros quinzenais nas instalações dos laboratórios de desenvolvimento de software localizados no campus Vilhena.

O processo de desenvolvimento foi conduzido por meio da utilização dos recursos oferecidos pela ferramenta GitLab. Esta plataforma permitiu a criação de quadros nos quais foram definidas as funcionalidades a serem desenvolvidas, bem como aquelas que estavam em progresso e as que já haviam sido concluídas. A figura a seguir ilustra o quadro de gestão adotado para o projeto.

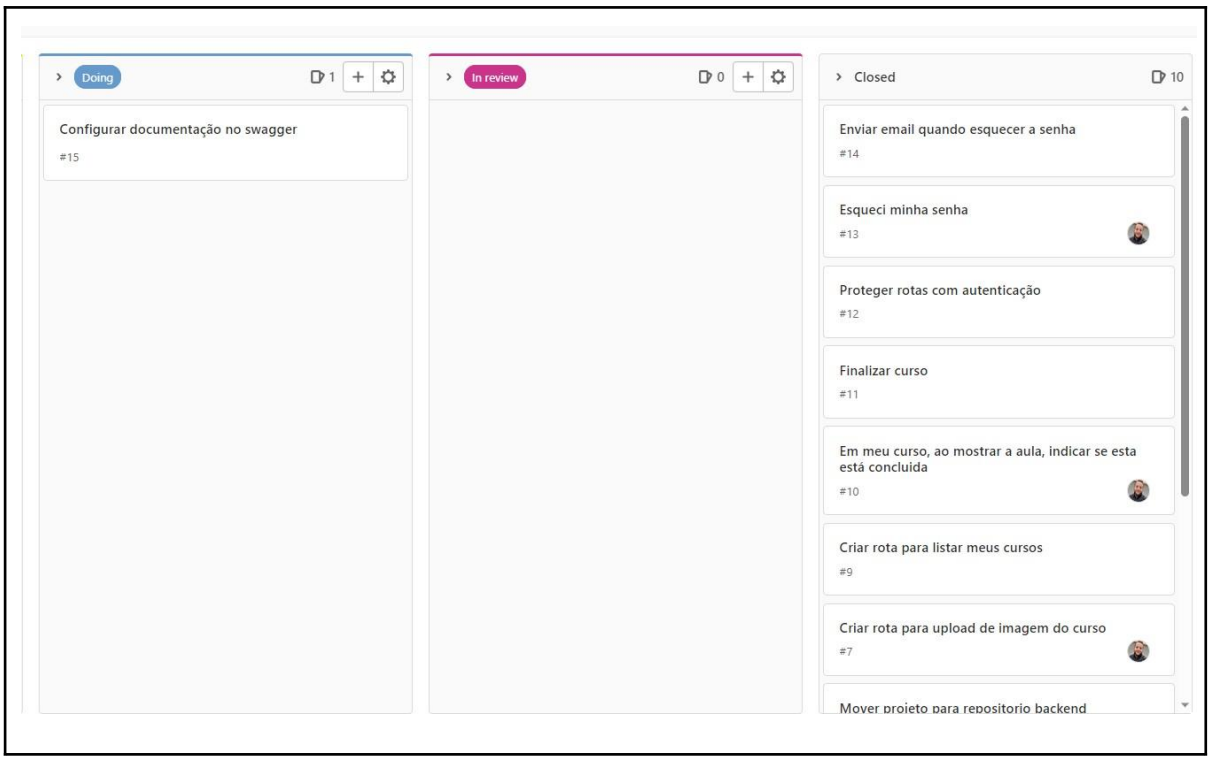

Figura 1. Exemplo de board no gitlab

A definição dos requisitos ocorreu em colaboração com o professor orientador e foi delineada por meio de reuniões estruturadas. Durante esses encontros, foi possível esclarecer as funcionalidades essenciais do sistema. Além disso, foram debatidas abordagens para a construção da arquitetura do banco de dados, bem como a maneira pela qual o cliente esperava que o layout do sistema fosse apresentado.

As definições dos requisitos funcionais desempenharam um papel crucial na delimitação do escopo do projeto, permitindo-nos concentrar nossos esforços na criação de um Produto Mínimo Viável (PMV) e garantindo a entrega dentro dos prazos estipulados. Por outro lado, as definições dos requisitos não funcionais atuaram como guia para o desenvolvimento das competências adquiridas ao longo do curso, além de orientar a produção de um código que aderisse a boas práticas de programação.

A elaboração do backend foi executada empregando a técnica de Mapeamento Objeto-Relacional (ORM) por meio da ferramenta Sequelize. Isso permitiu estabelecer uma conexão eficaz com o banco de dados, agilizando o processo de comunicação e proporcionando uma estrutura bem organizada. Adicionalmente, o ORM possibilitou a administração simplificada das migrações e o

controle de versionamento. Esta ferramenta ofereceu recursos como a geração e sincronização automatizada do banco de dados através da elaboração de classes em JavaScript, acompanhada da utilização de funções para a gestão do acesso à base de dados.

A consideração pela segurança da informação permeou todo o nosso projeto, destacando-se como um pilar fundamental. Acentuamos a nossa preocupação em salvaguardar informações sensíveis e prevenir possíveis vazamentos, resultando na implementação de um sistema de controle de acesso robusto. Esse mecanismo foi concretizado por meio da utilização de credenciais, as quais desempenharam um papel crucial na validação da identidade dos usuários.

A abordagem que adotamos incluiu a geração de tokens, que atuam como "chaves digitais" para permitir o acesso autorizado ao sistema. A fim de implementar o processo de autenticação, optamos pelo uso do JSON Web Token (JWT), uma tecnologia que nos permitiu criptografar as informações de autenticação, garantindo a segurança da transmissão desses dados. Adicionalmente, o JWT viabilizou a verificação da autenticidade das credenciais apresentadas.

Com essa abordagem, pudemos assegurar que somente usuários autorizados pudessem acessar a plataforma, reforçando o compromisso com a confidencialidade e a integridade dos dados.

No processo de desenvolvimento da API, adotamos o emprego do framework Express, que se mostrou uma escolha sólida e eficaz. Por meio deste framework, conseguimos criar um serviço web que se incumbiu da administração dos dados presentes no banco de dados, bem como da gestão das rotas e da disponibilização das informações necessárias.

Para garantir a integridade e validade dos dados, incorporamos a dependência express-validator, a qual integra o ecossistema do Node.js. Essa ferramenta se destacou pela sua capacidade de efetuar a validação precisa das entradas de dados e realizar correções nos parâmetros fornecidos nas rotas da API. Isso não somente aprimorou a confiabilidade dos dados manipulados, mas também contribuiu para a prevenção de erros e inconsistências.

Em conjunto, o uso do framework Express e da dependência express-validator viabilizou o desenvolvimento de uma API coesa, confiável e de alto desempenho, que atendeu aos requisitos de gerenciamento de dados, controle de rotas e garantia da qualidade das informações.

O backend da aplicação foi projetado e documentado utilizando a ferramenta Swagger. Com essa abordagem, conseguimos definir e parametrizar as várias rotas presentes no sistema de forma organizada. Isso nos proporcionou uma compreensão clara de como as diferentes partes do backend interagem e quais informações são necessárias em cada interação.

No que diz respeito ao frontend, optamos por utilizar as ferramentas fornecidas pelo Next.js. Através da criação de componentes estruturados, desenvolvemos páginas que trocam informações de maneira fluída entre si. Essa abordagem permitiu a construção de uma interface de usuário coesa e responsiva, simplificando a troca de informações e a navegação pelo sistema.

Ao empregar o Swagger para a documentação e planejamento do backend, e aproveitar as capacidades do Next.js no desenvolvimento do frontend, conseguimos criar uma aplicação integrada na qual os componentes do sistema interagem de maneira eficaz.

Para quantificar os resultados do projeto, estabelecemos critérios específicos. Delimitamos a criação de um conjunto de 3 páginas cruciais para o sistema, além de determinar o número de rotas que seria documentado no Swagger.

Conseguimos concretizar a entrega das páginas front-end conforme proposto no escopo do projeto, juntamente com a criação de aproximadamente 38 rotas no backend, o que reflete a funcionalidade abrangente do sistema.

Para estabelecer e delinear o cronograma do projeto, recorremos a um gráfico de Gantt, uma ferramenta que possibilitou prever a implementação do código do projeto e avaliar o tempo necessário para o desenvolvimento do sistema. Essa representação visual está ilustrada na figura abaixo:

Figura 2. Gráfico Gantt do projeto

Inicialmente, a conclusão dentro do prazo estabelecido enfrentou obstáculos de natureza diversa, principalmente em relação à dinâmica da equipe e à necessidade de reavaliação das prioridades definidas. No entanto, em resposta a esses desafios, foi determinado um novo cronograma e âmbito do projeto, permitindo a formulação de um plano que viabilizasse a entrega de um projeto realizável até o término do curso.

2 PRODUTO

2.1 Arquitetura da aplicação

A arquitetura da aplicação, também conhecida como arquitetura de software, refere-se à estrutura organizacional e funcional do sistema de software que suporta a plataforma de cursos MOOC. Ela define como os diferentes componentes da plataforma, como a interface do usuário, o banco de dados, os serviços de autenticação e os módulos de curso, estão interconectados. A arquitetura da aplicação MOOC é projetada para acomodar os requisitos específicos desse ambiente de educação online, como a oferta de diversos cursos, gerenciamento de conteúdo, interações entre alunos e instrutores, avaliações e recursos multimídia,

garantindo ao mesmo tempo a usabilidade, desempenho e segurança da plataforma. Para essa aplicação, utilizamos a abordagem C4 Model (Modelo C4).

2.2 C4 Model

A abordagem C4 Model ajuda a entender de forma sistemática e clara a representação da arquitetura de software de forma escalável e compreensível. Essa abordagem, possibilita às equipes comunicarem de forma eficaz a estrutura se seus sistemas (Brown, 2018).

O Modelo C4 é dividido em 5 partes: Contexto, Contêineres, Componentes e Código. A seguir, vamos apresentar as três primeiras partes do Modelo C4 no contexto da aplicação MOOC.

2.2.1 Contexto

A camada de contexto (figura 3), demonstra a visão mais ampla da plataforma MOOC, identificando suas interações com atores externos, como alunos, instrutores e administradores. Nessa camada vemos também a interação com a plataforma YouTube, onde ficarão armazenados as vídeo aulas. Essa representação visual apresenta as fronteiras e limites da plataforma, fornecendo um panorama geral do sistema.

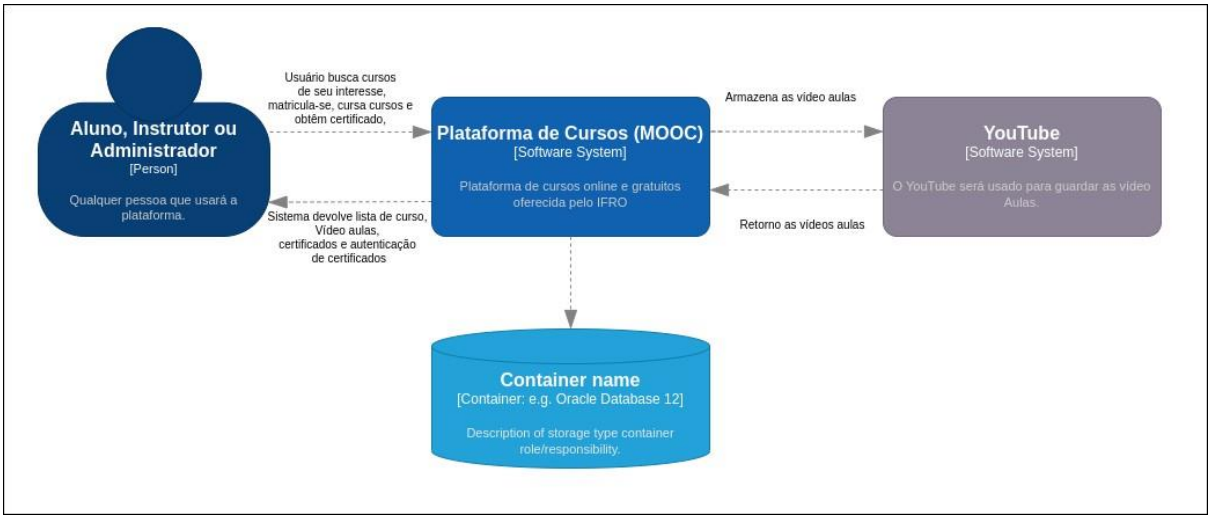

Figura 3. Modelo C4 - Contexto

2.2.2 Contêineres

Na parte de contêineres (figura 4), os principais componentes da plataforma MOOC são identificados, incluindo aplicativos web, bancos de dados, serviços de comunicação e módulos de processamento. Essa camada descreve como esses contêineres colaboram para entregar as funcionalidades da plataforma.

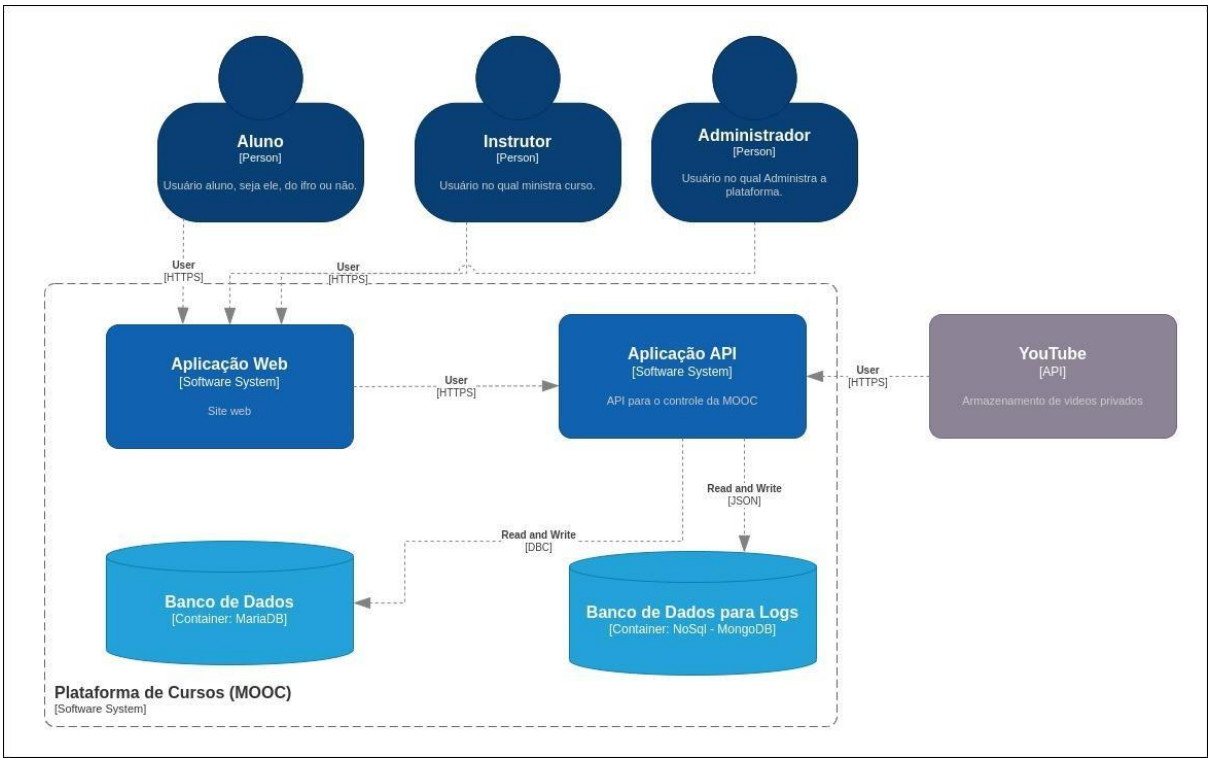

Figura 4. Model C4 - Container

2.2.3 Componentes

Na fase de "Componentes" do Modelo C4, a arquitetura da aplicação MOOC é apresentada em detalhes mais profundos, proporcionando uma visão ampliada dos principais elementos funcionais que compõem cada contêiner identificado anteriormente. Nesse estágio, o foco está sobre os módulos e componentes específicos que desempenham papéis essenciais na operação da plataforma de ensino online.

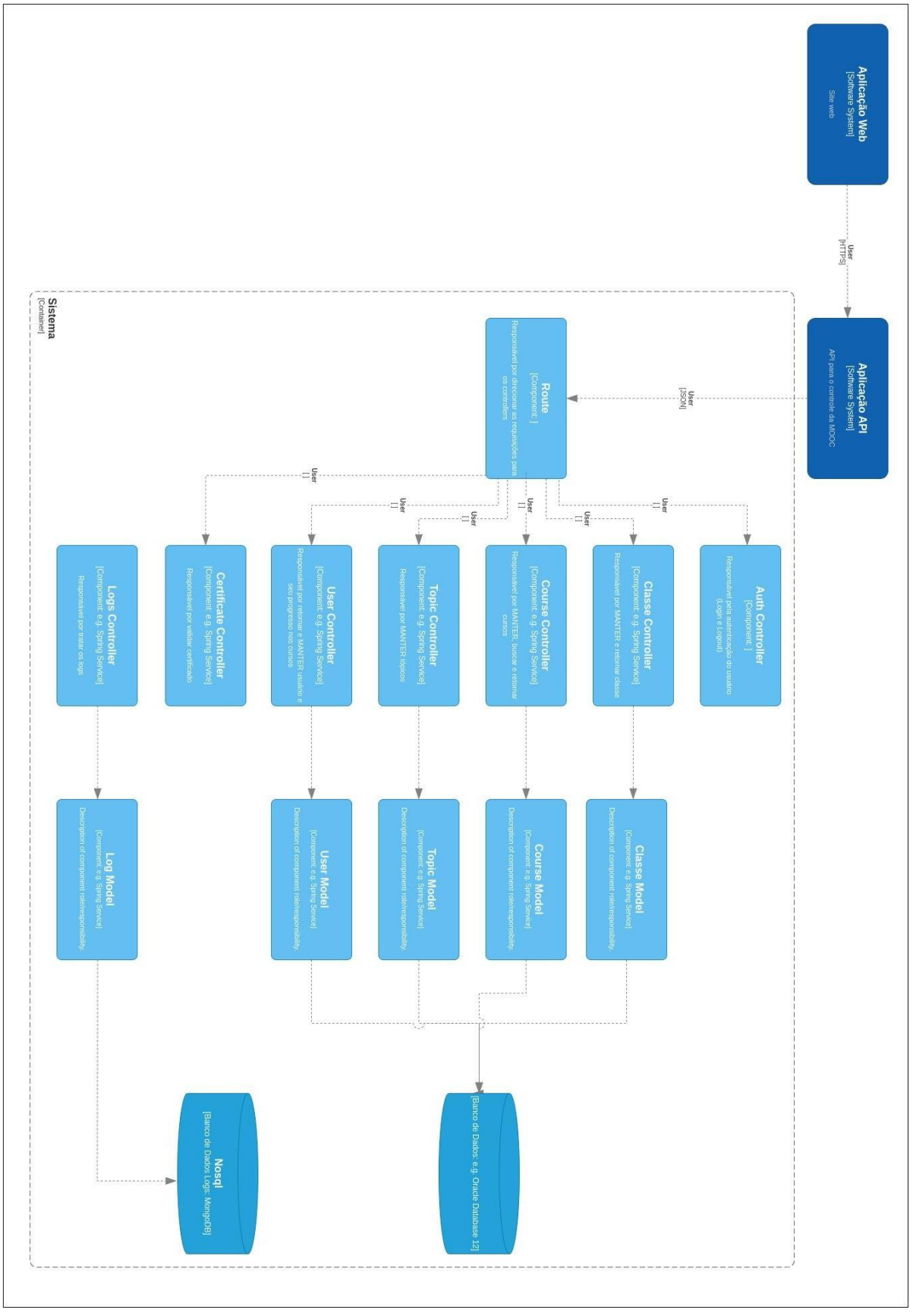

Figura 5. Modelo C4 - Componentes

A arquitetura da API foi desenvolvida de maneira monolítica, o que significa que todos os serviços da API estão integrados em uma única estrutura. Inicialmente, o fluxo da requisição começa no arquivo index.js, que, por sua vez, encaminha a solicitação para o arquivo routes.js. Nesse processo, as funcionalidades de segurança e validação entram em ação, garantindo a verificação dos dados de entrada. Posteriormente, a requisição é direcionada aos controladores (controllers), responsáveis pelo processamento inicial e pelo encaminhamento das ações para os modelos (models). Os modelos, por sua vez, estabelecem a conexão com o banco de dados e executam as operações necessárias, retornando os resultados ao controlador correspondente. Por fim, os controladores concluem o processamento da requisição e enviam as respostas finais ao usuário. A Figura 5, apresenta uma representação visual dessa arquitetura.

2.2 Modelo de dados

A modelagem de dados da plataforma MOOC é minuciosamente estruturada e interconectada, com o objetivo de representar eficazmente as diversas entidades e suas interações no ambiente educacional online. Através da utilização da biblioteca Sequelize, que abstrai os comandos de operações de SQL, e faz com que possamos usar linguagem de programação que já estamos usando no backend para nos conectarmos e operarmos o banco em uma aplicação Node.js, as entidades cruciais são definidas, permitindo uma organização lógica e coesa das informações. Através do uso desse modelo de dados, é possível aprimorar a administração das informações, possibilitando análises detalhadas do desempenho dos alunos, a customização da experiência de aprendizado e a implementação de funcionalidades como a emissão de certificados de conclusão (AMOASEI, 2021).

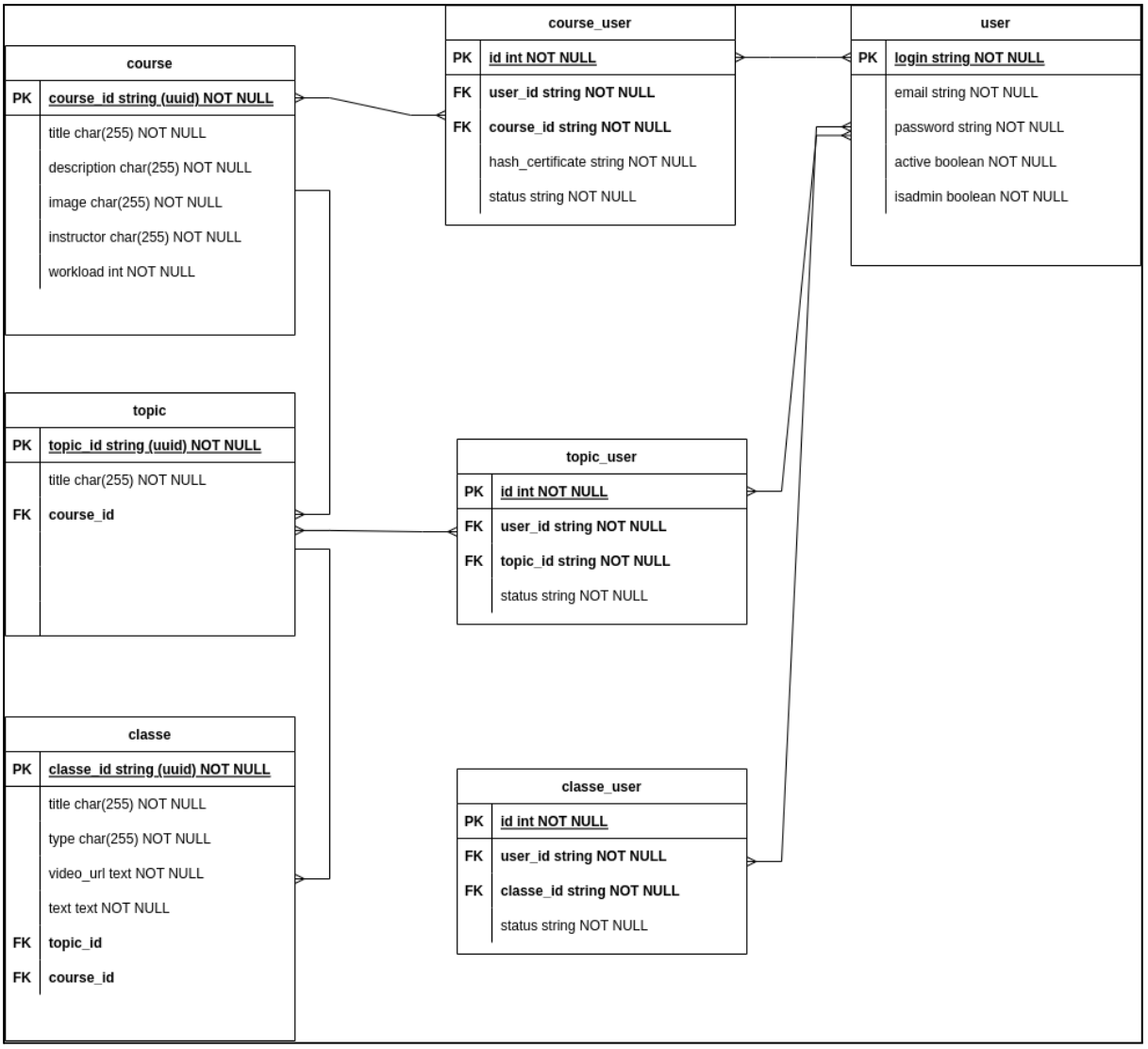

Figura 6 - Diagrama Entidade de Relacionamento

A entidade “User” contém informações detalhadas importantes, englobando dados de login, email, password, active e isadmin. Isso viabiliza a autenticação, autorização e monitoramento das atividades dos usuários na plataforma.

As entidades “Userclasse”, “UserTopic”, “UserCourse” são denominadas de intermediárias a qual se tem uma relação de muitos-para-muitos. Ficando registro em cada classe intermediária a interação de cada usuário em relação às classes, tópicos e cursos ali escolhido. As associações “belongsToMany”, refletem as interações genuínas que ocorrem na plataforma, essas associações habilitam o acompanhamento da participação dos usuários, fornecendo informações valiosas para análises de desempenho, personalização de conteúdo e monitoramento individualizado (CAKEPHP, 2020).

user.model.js 2.11 KB

```
const { DataTypes } = require('sequelize');
const database = require('../config/database.config.js');

const Classe = require('./classe.model.js');
const Course = require('./course.model.js');
const Topic = require('./topic.model.js');


const User = database.define('users', {
    login: {
        type: DataTypes.STRING(50),
        primaryKey: true
    },
    email: {
        type: DataTypes.STRING(100),
        allowNull: false,
        unique: true
    },
    password: {
        type: DataTypes.STRING(255),
        allowNull: false
    },
    active: {
        type: DataTypes.BOOLEAN,
        allowNull: false
    },
    isadmin: {
        type: DataTypes.BOOLEAN,
        allowNull: false
    }
});

const UserClasse = database.define('user_classes',{
    status: DataTypes.CHAR,
    user_login : {
        type: DataTypes.STRING,
        primaryKey: false,
        references: {
            model: User,
            key: "login"
        }
    },
    classe_id : {
        type: DataTypes.STRING,
        primaryKey: false,
        references: {
            model: Classe,
            key: "id"
        }
    }
})

const UserTopic = database.define('user_topics',{
    status: DataTypes.CHAR,
    user_login : {
        type: DataTypes.STRING,
        primaryKey: false,
        references: {
            model: User,
            key: "login"
```

Figura 7. user.model.js - sequelize

```
        }
    },
    topic_id : {
        type: DataTypes.STRING,
        primaryKey: false,
        references: {
            model: Topic,
            key: "id"
        }
    }
})

const UserCourse = database.define('user_courses',{
    status: DataTypes.CHAR,
    user_login : {
        type: DataTypes.STRING,
    },
    course_id : {
        type: DataTypes.STRING,
    }
})

User.belongsToMany(Classe, {
    through: 'user_classes',
})

Classe.belongsToMany(User, {
    through: 'user_classes',
})

User.belongsToMany(Course, {
    through: 'user_courses',
})

Course.belongsToMany(User, {
    through: 'user_courses',
})

User.belongsToMany(Topic, {
    through: 'user_topics',
})

Topic.belongsToMany(User, {
    through: 'user_topics',
})


module.exports = {User, UserClasse, UserCourse, UserTopic};
```

Figura 8. user.model.js - sequelize

2.3 Documentação da API

MOOC é uma plataforma de cursos online para alunos com conteúdo de diferentes materiais e assuntos. Poderão acompanhar progresso, históricos e ter acesso a certificado ao final de cada curso concluído. O acesso à plataforma poderá ser de forma pública ou autenticado.

Base URL

A URL base para todas a solicitações é:

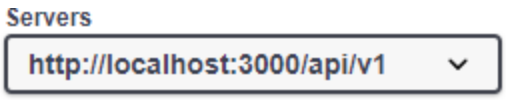

Figura 9. Swagger - Base URL

Categorias e Endpoints:

Auth - Tudo sobre autenticação

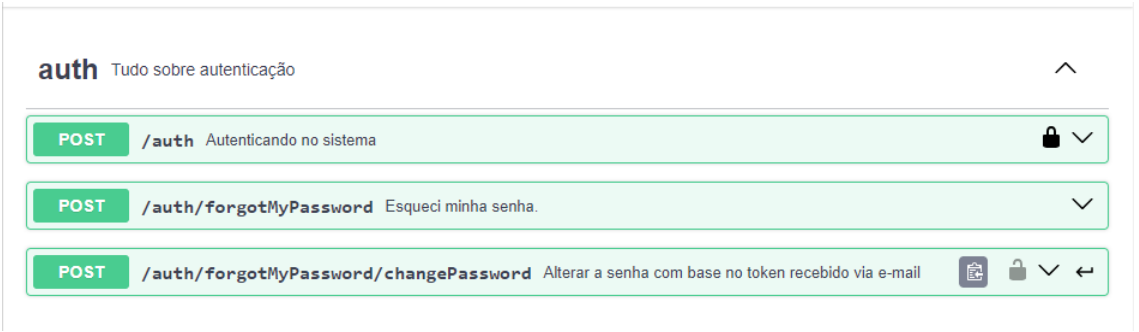

Figura 10. Swagger - Autenticação do sistema

Descrição: Autenticado no sistema, esqueci minha senha e alterar a senha com base no token recebido via e-mail

Course - Tudo sobre cursos

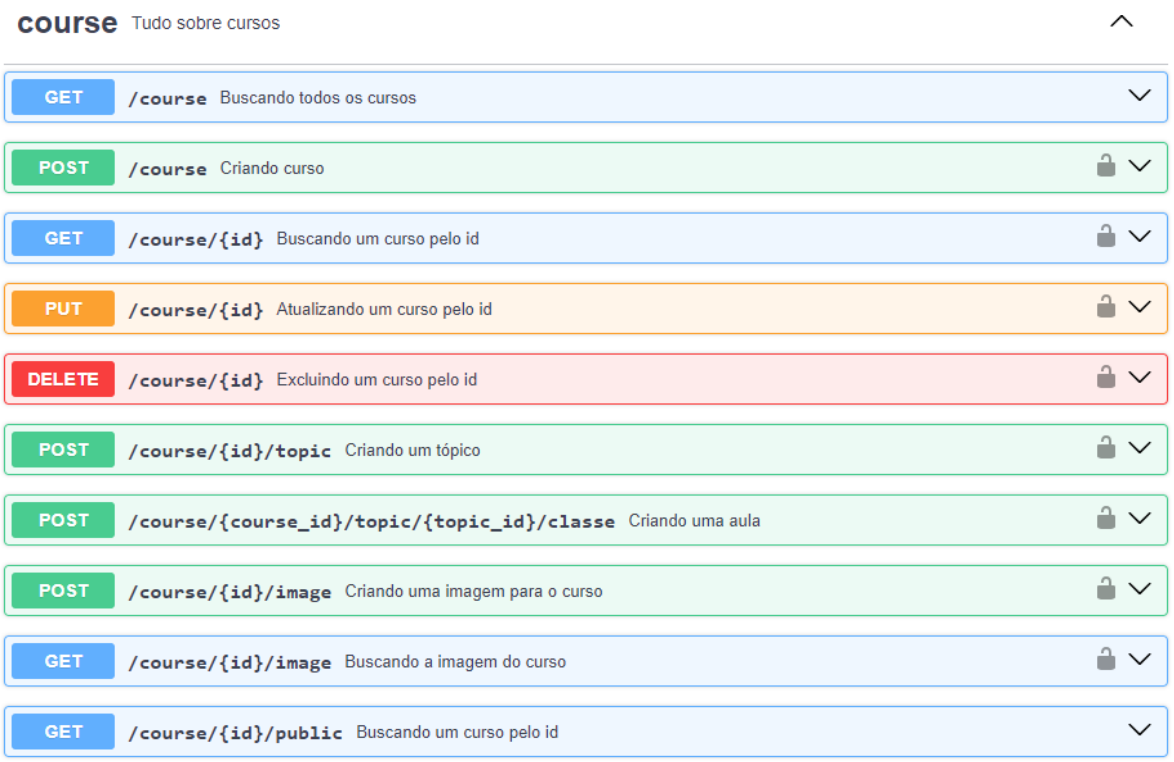

Figura 11. Swagger - Tudo sobre cursos

Descrição: Buscando todos os cursos, criando curso, buscando um curso pelo id, atualizando um curso pelo id, excluindo um curso pelo id, criando um tópico, criando uma aula, criando uma imagem para o curso, buscando a imagem do curso, buscando um curso pelo id.

Topics - Tudo sobre tópicos

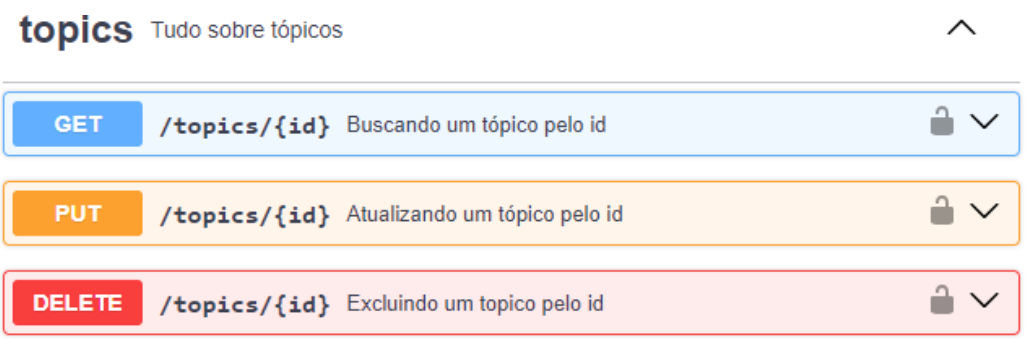

Figura 12. Swagger - Tudo sobre tópicos

Descrição: Buscando um tópico pelo id, atualizando um tópico pelo id, excluindo um tópico pelo id.

Classes - Tudo sobre classes

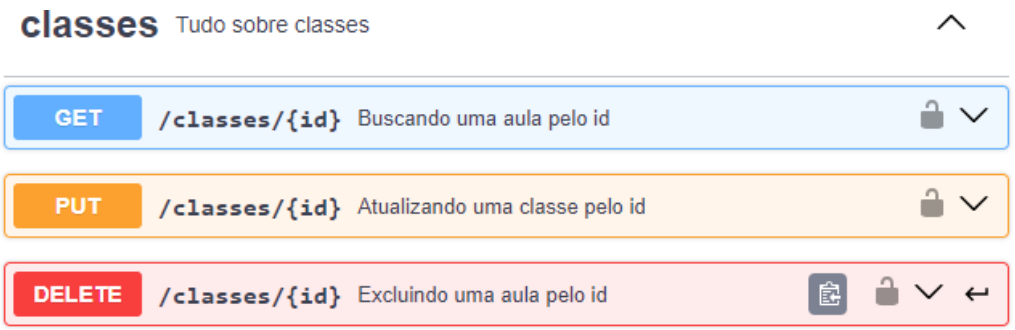

Figura 13. Swagger - Tudo sobre classes

Descrição: Buscando uma aula pelo id, atualizando uma classe pelo id, excluindo uma aula pelo id.

User - Tudo sobre usuários

Figura 14. Swagger - Tudo sobre usuários

Descrição: Criando usuário, buscando todos os usuários, buscando um usuário pelo id, atualizando um usuário pelo id, excluindo um usuário pelo id, buscando o certificado pelo id do usuário e pelo id do certificado, busca dados sobre o curso do usuário, buscando todos os meus cursos.

User course - Tudo sobre usuários e cursos

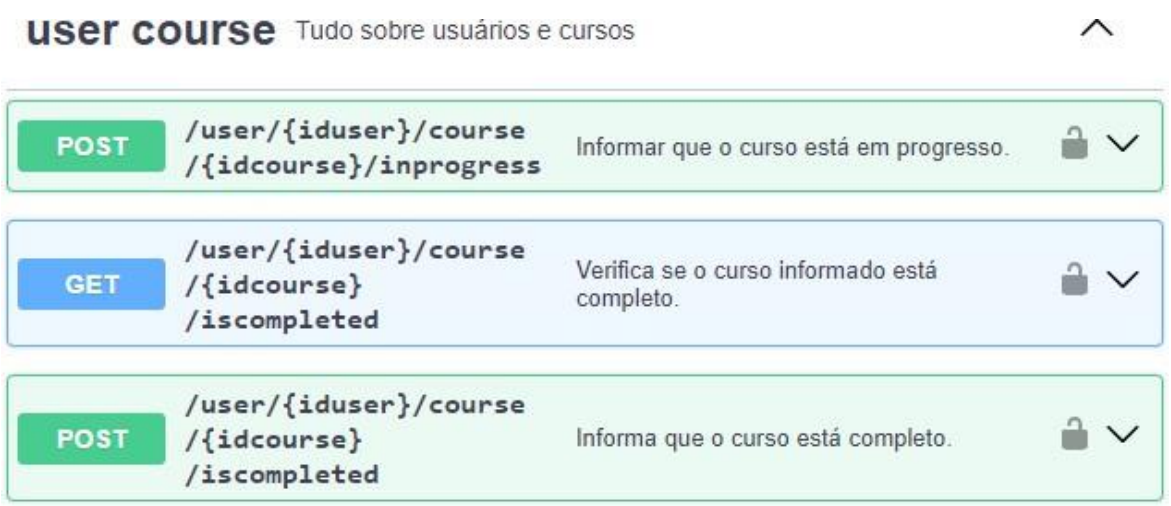

Figura 15. Swagger - Tudo sobre usuários e cursos

Descrição: Informar que o curso está em progresso, verificar se o curso informado está completo, informar que o curso está completo.

User topic - Tudo sobre usuários e tópicos

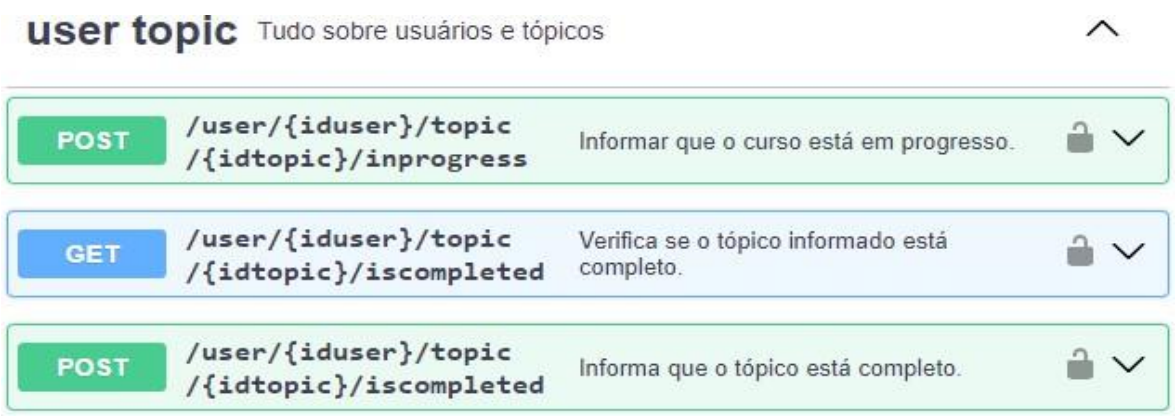

Figura 16. Swagger - Tudo sobre usuários e tópicos

Descrição: Informar que o curso está em progresso, verifica se o tópico informado está completo, informa que o tópico está completo.

User classe - Tudo sobre usuários e aulas

User classe - Tudo sobre usuários e aulas

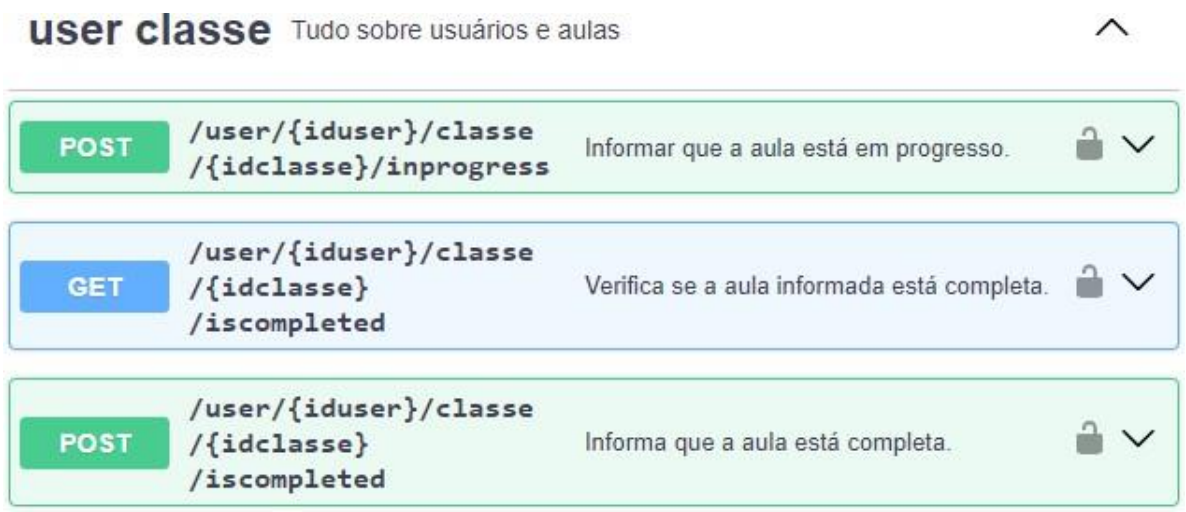

Figura 17. Swagger - Tudo sobre usuários e aulas

Descrição: Informar que a aula está em progresso, verifica se a aula informada está completa, informa que a aula está completa.

Certificate - Tudo sobre certificado

Figura 18. Swagger - Tudo sobre certificado

Descrição: Validar hash do certificado.

2.4 Screenshots

A aplicação desenvolvida foi, basicamente, separada em duas partes: Front-end e Back-end. A seguir será apresentado as screenshots, elas oferecem um vislumbre da experiência educacional transformadora oferecida pela plataforma de cursos MOOC do IFRO (Instituto Federal de Educação, Ciência e Tecnologia de Rondônia).

A figura 18 demonstra a página inicial da aplicação, nela é listado todos os cursos disponíveis. O usuário pode realizar a busca através do nome do curso interessado.

Figura 19. Página Inicial

A figura 19 apresenta a página onde o usuário obtém mais informações sobre o curso desejado, como por exemplo, tópicos, instrutor e carga horária.

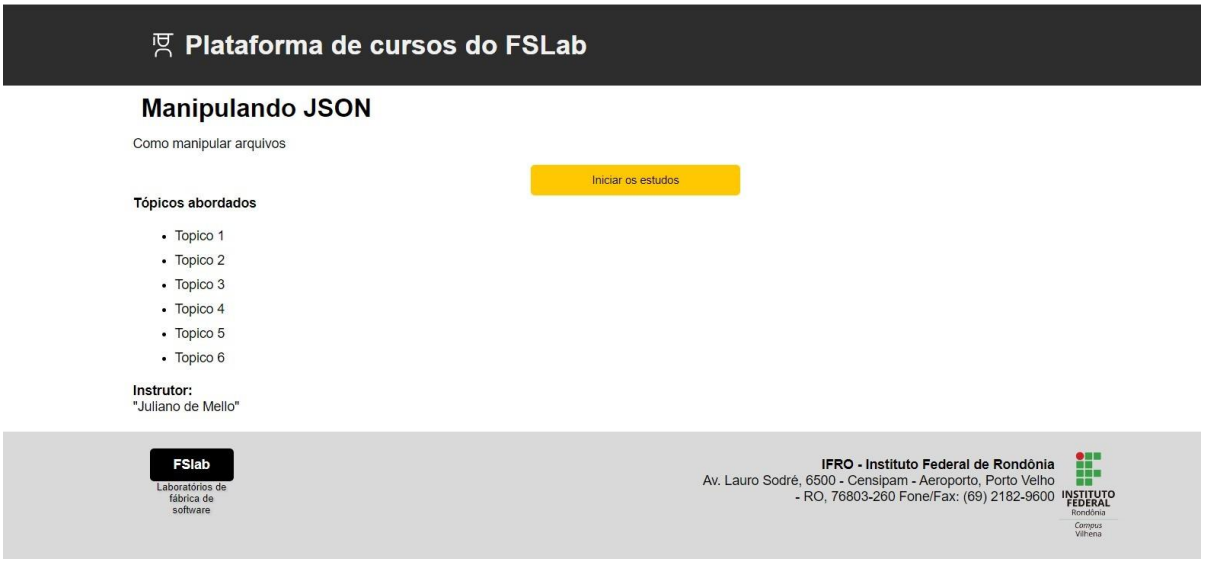

Figura 20. Página detalhe do curso

A figura 20 retrata a página onde o usuário irá assistir as aulas e acompanhar o seu progresso dentro do curso. Atualmente, essa página oferece apenas as duas funcionalidades citadas, no entanto, há planos futuros de adicionar as funcionalidades de aulas em texto, atividades extras e testes avaliativos.

Figura 21. Página video aulas

3. CONCLUSÃO

Neste projeto desafiador de criação de uma plataforma de cursos online para o Instituto Federal de Rondônia (IFRO), a equipe composta por Fabrício, Erick, Lucas e Juliano demonstrou sua capacidade de desenvolvimento de software de alto desempenho e qualidade excepcional. Sob a orientação do professor Marcos, esta jornada foi conduzida com êxito, com cada membro da equipe desempenhando um papel vital na concretização dessa visão.

O projeto começou com a definição cuidadosa dos requisitos, alcançada através de colaboração ativa com o professor orientador e reuniões estruturadas. Essas sessões permitiram esclarecer as funcionalidades essenciais, discutir a arquitetura do banco de dados e delinear o layout da plataforma. Os requisitos funcionais guiaram a delimitação do escopo do projeto, focando na criação de um Produto Mínimo Viável (PMV) e garantindo que os prazos fossem cumpridos. Os requisitos não funcionais, por outro lado, foram fundamentais para garantir a qualidade técnica do código e a eficiência do sistema.

O desenvolvimento do backend e das APIs, liderado por Juliano, foi marcado pela escolha do framework Express e pela integração da dependência express-validator. Essas decisões resultaram em uma API robusta que gerencia de forma eficaz os dados do sistema e controla as rotas com precisão. A documentação detalhada utilizando a ferramenta Swagger proporcionou uma compreensão clara de como as partes do backend interagiam, simplificando o desenvolvimento e a manutenção.

A segurança da informação, um pilar fundamental do projeto, foi enfatizada em todo o processo. Para garantir a confidencialidade e a integridade dos dados, implementamos um sistema de controle de acesso robusto com a orientação do professor Marcos. A equipe introduziu tokens JWT como "chaves digitais", criptografando as informações de autenticação e garantindo a segurança da transmissão de dados. Isso assegurou que apenas usuários autorizados pudessem acessar a plataforma, reforçando nosso compromisso com a proteção dos dados.

No frontend, a equipe optou pelo Next.js para criar uma interface de usuário coesa e responsiva. A construção de componentes estruturados permitiu a troca de informações de maneira fluida, simplificando a navegação pelo sistema. Essa

abordagem resultou em uma experiência amigável para os usuários finais, promovendo a acessibilidade e a usabilidade do site

A quantificação dos resultados do projeto foi alcançada através da entrega das páginas front-end conforme proposto no escopo do projeto, juntamente com a criação de aproximadamente 38 rotas no backend, demonstrando a funcionalidade abrangente do sistema. Além disso, o uso de um gráfico de Gantt permite um gerenciamento eficaz do cronograma, garantindo a entrega dentro dos prazos estipulados.

Em resumo, o projeto desenvolvido sob a orientação do professor Marcos para o Instituto Federal de Rondônia (IFRO) é um exemplo notável de como a colaboração na definição de requisitos, a escolha criteriosa de tecnologias e ferramentas, o foco na segurança da informação e a documentação cuidadosa podem resultar em uma plataforma de cursos online eficaz e de alta qualidade. Esta equipe demonstrou sua dedicação à excelência no desenvolvimento de software e como esses princípios podem ser aplicados com sucesso em projetos reais.

3 Referências

AMOASEI, Juliana. **ORM com NodeJS: API com Sequelize e MySQL.** Disponivel em:<https://www.alura.com.br/conteudo/orm-nodejs-api-sequelize-mysql#:~:text=O%20Sequelize%20abstrai%20os%20comandos,que%20est%C3%A1%20escrita%20em%20JavaScript.>. Acesso: 02 set. 2023.

Associações - Conectando tabelas. **CakePHP,** 2020. Disponivel em: <https://book.cakephp.org/3/pt/orm/associations.html#:~:text=BelongsToMany%20is%20muitas%20vezes%20referido,associa%C3%A7%C3%A3o%20BelongsToMany%20n%C3%A3o%20s%C3%A3o%20exclusivos.>. Acesso: 02 set. 2023.

Brown, Simon.O modelo C4 de documentação para Arquitetura de Software 2018. [Online; acessado 06-Setembro-2022].
Disponível em: <https://www.infoq.com/br/articles/C4-architecture-model/>. Citado na página 7.