Politécnico
de Viseu

Escola Superior
de Tecnologia
e Gestão de Viseu

departamento de
informática

Aplicações para a Internet III
Trabalho Prático

Cassandra Sousa Veríssimo pv27077

Daniela Skachko pv25656

Dezembro 2026

Índice

Introdução

O presente projeto tem como objetivo o desenvolvimento de um site académico destinado à partilha de recursos educativos entre estudantes. O sistema permite o acesso a diferentes pastas e ficheiros associados a disciplinas, facilitando a consulta e organização de materiais relevantes para o estudo. Embora todos os utilizadores possam visualizar os ficheiros disponíveis na plataforma, apenas os alunos registados têm permissão para realizar o download e avaliar os conteúdos, promovendo assim um ambiente participativo e colaborativo.

O desenvolvimento do projeto teve início no Swagger Editor, onde foi estruturada a base inicial e posteriormente ajustada de acordo com as correções sugeridas pelo professor. Numa fase posterior, o trabalho prosseguiu no Visual Studio, onde foram implementados os endpoints definidos no Swagger, configurada a base de dados externa em MongoDB e desenvolvidos o frontend e o backend do site.

Diagrama de contexto e diagrama de container

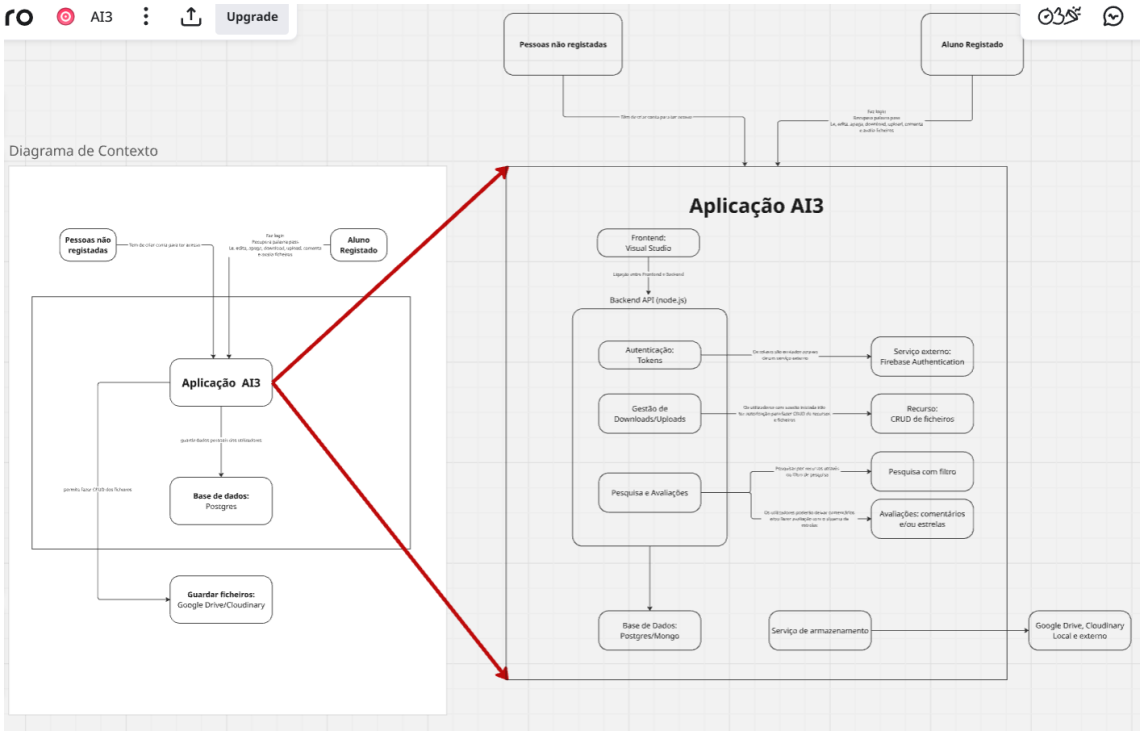

Figure 1 - Diagrama de contexto e diagrama de container

Nos diagramas apresentados, é possível observar a estrutura base da construção do projeto desenvolvida na plataforma Miro. A representação inicia-se pela identificação dos dois tipos de utilizadores, demonstrando os respetivos direitos e permissões no sistema e partir desta distinção, o diagrama evolui para a secção dedicada ao funcionamento global do site.

Nesta caixa encontram-se detalhadas as diversas ações que o site permite realizar, acompanhadas de descrições que explicam o papel e o funcionamento de cada uma. Esta organização facilita a compreensão da lógica interna do sistema e a forma como cada funcionalidade contribui para o objetivo principal da plataforma.

O diagrama inclui ainda a indicação das possíveis bases de dados externas que poderão ser utilizadas, por exemplo o Postgres ou MongoDB, destacando a sua importância no armazenamento dos ficheiros. Além disso, é apresentado o serviço responsável pela guarda dos recursos, como Google Drive ou Cloudinary, assegurando

que os ficheiros permanecem acessíveis, organizados e integrados com o restante sistema.

Swagger Editor

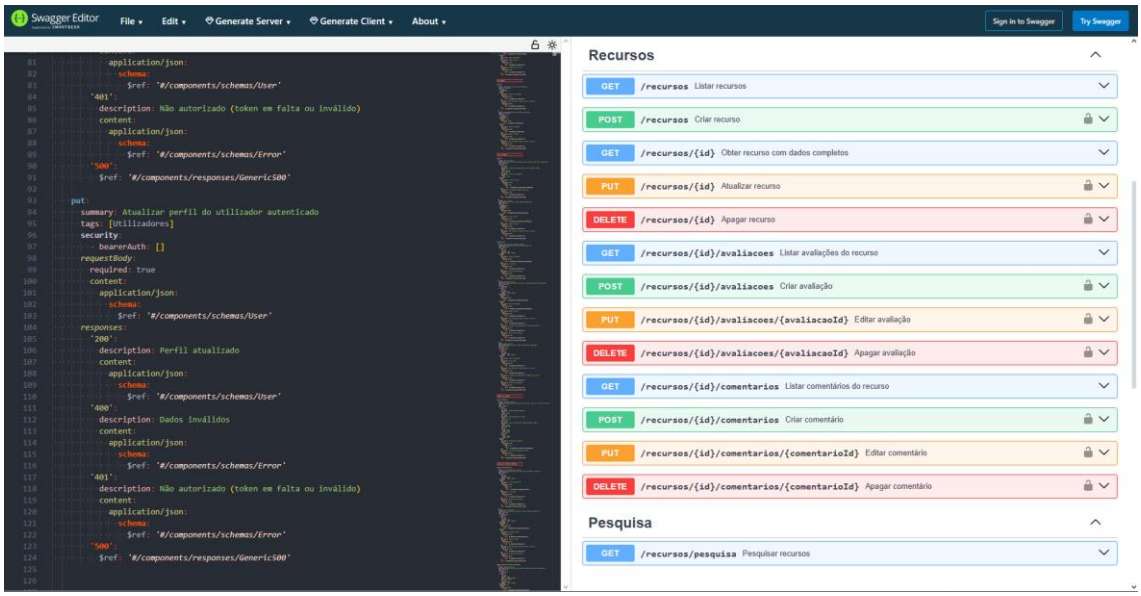

Figura 2 - Swagger Editor

A segunda fase do desenvolvimento consistiu na edição e atualização contínua do Swagger, uma etapa fundamental para garantir a coerência entre a documentação da API e a implementação prática realizada no Visual Studio. À medida que novas funcionalidades eram adicionadas ao projeto, procedíamos simultaneamente às respetivas modificações no Swagger Editor.

Durante este processo, foram documentados todos os endpoints existentes, incluindo os seus métodos, parâmetros, respostas esperadas e possíveis códigos de erro que o sistema poderia gerar, permitindo uma visão mais clara do funcionamento da plataforma.

MongoDB

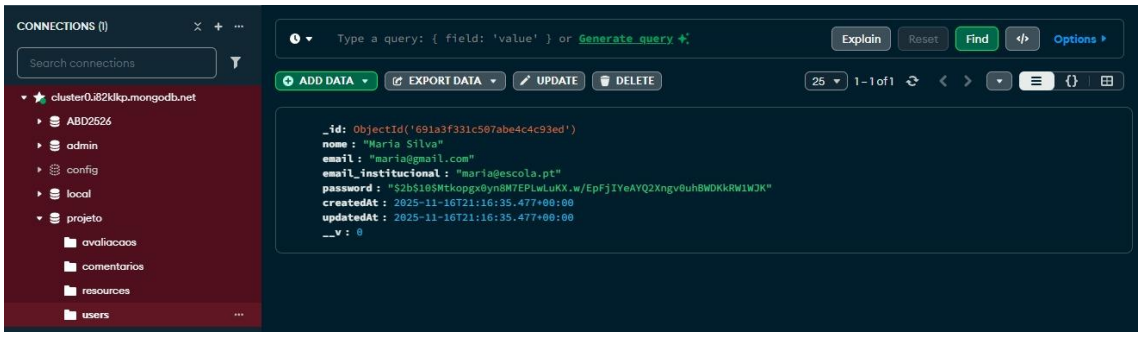

Figura 3 – MongoDB

A base de dados externa escolhida para o desenvolvimento da plataforma foi MongoDB, que permite fazer alterações futuras sem comprometer a integridade do sistema. Sempre que um utilizador efetua o seu registo na plataforma, todas as informações introduzidas são automaticamente enviadas e armazenadas no MongoDB, assegurando um processo protegido.

Os dados recolhidos incluem o id, nome, e-mail, email institucional, password, em que dia foi registado e em que dia foi modificado , elementos essenciais para garantir a sua identificação e autenticação no acesso às diferentes funcionalidades disponíveis no site.

Desenvolvimento do projeto

No desenvolvimento desta plataforma foi criada uma aplicação composta por um backend em Node.js/Express e um frontend em React, permitindo que diferentes utilizadores possam registar-se, autenticar-se e gerir recursos académicos, incluindo avaliações e comentários.

Para garantir segurança desde o início, implementou-se um sistema de autenticação com JWT, que permite ao servidor identificar o utilizador após o login. Sempre que um utilizador inicia sessão, o servidor valida as credenciais e gera um token seguro. Esse token é guardado no frontend e enviado automaticamente em todas as ações protegidas, garantindo que apenas utilizadores autenticados conseguem criar, editar ou apagar conteúdos.

Foi também implementado um mecanismo de controlo de permissões, assegurando que cada utilizador só pode editar ou eliminar os recursos, comentários ou avaliações que criou. Este controlo evita alterações indevidas ou acessos não autorizados, mesmo que alguém tente manipular pedidos através do browser ou de ferramentas externas.

Para reforçar ainda mais a segurança, integrou-se o sistema Google reCAPTCHA, utilizado tanto no registo como no login. O reCAPTCHA serve para impedir atividades automatizadas, garantindo que apenas utilizadores reais conseguem criar contas ou tentar iniciar sessão. O frontend recolhe o token gerado pelo reCAPTCHA e o backend valida-o diretamente com os servidores da Google antes de permitir o acesso.

Além disso, foram aplicadas outras medidas de proteção, como sanitização de inputs contra XSS, limitação de pedidos (rate limiting) para evitar ataques de força bruta e encriptação de passwords com bcrypt.

Conclusão

O desenvolvimento desta plataforma permitiu criar um sistema funcional e seguro para gestão e partilha de recursos académicos. Ao longo do projeto, foram aplicados princípios fundamentais de desenvolvimento web moderno: uma API REST estruturada em Node.js com Express, persistência de dados em MongoDB e um frontend em React que oferece uma interface simples e eficaz ao utilizador.

Do ponto de vista da segurança, foram integrados mecanismos essenciais como autenticação por JWT, validação de permissões para garantir que apenas o autor pode editar ou remover os seus próprios conteúdos, e a implementação do Google reCAPTCHA, que reforça a proteção do sistema contra tentativas de acesso automatizado. Estes elementos contribuíram para uma aplicação mais robusta e alinhada com boas práticas do mercado.

A documentação em OpenAPI permitiu formalizar todas as rotas e comportamentos da API, garantindo clareza e facilitando futuras extensões ou manutenção do sistema.

Em conjunto, todos estes componentes resultaram numa plataforma completa, escalável e segura, capaz de suportar a partilha responsável de recursos entre utilizadores e que poderá ser facilmente expandida com novas funcionalidades no futuro.