



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ**  
**CAMPUS PICOS**  
**CURSO TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**KAUÊ ALVES SILVA**

**REPOIF: DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA  
GERENCIAMENTO E COMPARTILHAMENTO CENTRALIZADO DE MATERIAIS  
DIDÁTICOS NO IFPI - CAMPUS PICOS**

**PICOS**  
**2026**

KAUÊ ALVES SILVA

REPOIF: DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA  
GERENCIAMENTO E COMPARTILHAMENTO CENTRALIZADO DE MATERIAIS  
DIDÁTICOS NO IFPI - CAMPUS PICOS

Trabalho de Conclusão de Curso apresentado ao  
Curso Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas do Instituto Federal  
de Educação, Ciência e Tecnologia do Piauí -  
Campus Picos, como requisito parcial para  
obtenção do título de Tecnólogo em Análise e  
Desenvolvimento de Sistemas.

Orientador: Prof. Me. Israel Cardoso Araujo.

PICOS

2026

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD**

Silva, Kauê Alves  
S586r      RepoIF: Desenvolvimento de uma plataforma web para gerenciamento e compartilhamento centralizado de materiais didáticos no IFPI - Campus Picos. / Kauê Alves Silva. - 2026.

103 f.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos, 2026.

Orientador: Prof. Me. Israel Cardoso Araujo.

1. Análise e Desenvolvimento de Sistemas. 2. Materiais Didáticos Digitais. 3. Repositórios Digitais. 4. Plataformas Web. 5. Usabilidade. I. Título.

CDD - 004.21

---

**Elaborado por Adriana Santos Magalhães – CRB 3/1857**

KAUÊ ALVES SILVA

REPOIF: DESENVOLVIMENTO DE UMA PLATAFORMA WEB PARA  
GERENCIAMENTO E COMPARTILHAMENTO CENTRALIZADO DE MATERIAIS  
DIDÁTICOS NO IFPI - CAMPUS PICOS

Trabalho de Conclusão de Curso apresentado ao  
Curso Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas do Instituto Federal  
de Educação, Ciência e Tecnologia do Piauí -  
Campus Picos, como requisito parcial para  
obtenção do título de Tecnólogo em Análise e  
Desenvolvimento de Sistemas.

Aprovado em: 07/08/2026.

BANCA EXAMINADORA

---

Prof. Me. Israel Cardoso Araujo (Orientador)  
Instituto Federal do Piauí (IFPI)

---

Prof. Me. João Paulo Lima do Nascimento  
Instituto Federal do Piauí (IFPI)

---

Prof. Esp. Raí Araújo de Miranda  
Universidade Federal do Piauí (UFPI)

Dedico este trabalho aos meus pais, aos meus amigos e aos mestres que, com apoio, companheirismo e ensinamentos, fizeram parte desta caminhada.

## **AGRADECIMENTOS**

A Deus, agradeço pela força, pela sabedoria e pela perseverança concedidas ao longo desta caminhada. Nos momentos de dificuldade e incerteza, a fé foi essencial para que eu continuasse seguindo em frente e encontrasse coragem para superar os desafios até a conclusão desta etapa.

Aos meus pais, agradeço pela confiança depositada em mim ao longo de toda essa trajetória, pela expectativa de sempre me ver alcançando novos objetivos e, especialmente, por todo o esforço realizado para que eu pudesse chegar até aqui. O apoio e os sacrifícios de vocês foram fundamentais para minha formação e para a conclusão deste trabalho.

À minha namorada, Alex, agradeço por ter me acompanhado durante todos os momentos do curso, compartilhando comigo as dificuldades, as conquistas e os desafios que fizeram parte dessa caminhada. Sua presença, seu apoio e suas brincadeiras tornaram esse percurso mais leve e significativo.

Aos meus amigos mais próximos, agradeço pela amizade, pelo companheirismo e pelos momentos compartilhados ao longo desses anos. Em especial, a Artur e Luis, por estarem ao meu lado desde o início dessa jornada, acompanhando de perto tantas etapas importantes. A Gabriel e Ana Lívia, por tornarem os dias de trabalho mais suportáveis, proporcionando conversas, risadas e momentos que fizeram toda a diferença durante essa caminhada.

Ao meu orientador, Prof. Me. Israel Cardoso Araujo, agradeço pela confiança no meu trabalho, pela orientação durante o desenvolvimento deste projeto e por todo o conhecimento compartilhado. Sua disponibilidade e suas contribuições foram importantes para o amadurecimento deste trabalho e para minha formação acadêmica.

Por fim, agradeço a todos que contribuíram para esta trajetória. A conclusão deste trabalho representa o encerramento de uma etapa marcada por aprendizado, desafios, crescimento e experiências compartilhadas.

*“Poderá vir um dia em que a coragem dos homens termine, quando desertarmos  
nossos amigos e trairmos os laços de amizade. Mas este dia não é hoje!”*

*Aragorn, O Senhor dos Anéis: O Retorno do Rei (2003).*

## RESUMO

A dispersão de materiais didáticos entre diferentes canais digitais pode dificultar sua organização, localização e recuperação no contexto acadêmico. Diante desse cenário, este trabalho teve como objetivo desenvolver e validar tecnicamente o RepolF, uma plataforma web voltada ao gerenciamento e ao compartilhamento centralizado de materiais didáticos no Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI) - Campus Picos. A solução foi desenvolvida para contemplar o gerenciamento de disciplinas, materiais, turmas e atividades, além de mecanismos de autenticação, autorização, controle de acesso, administração e moderação. A pesquisa caracteriza-se como aplicada, de abordagem qualitativa e objetivos exploratórios e descritivos. O desenvolvimento ocorreu de forma incremental, envolvendo levantamento e especificação de requisitos, modelagem, definição da arquitetura, implementação e validação. A solução utiliza React no frontend, NestJS no backend, PostgreSQL para persistência dos dados estruturados e Cloudflare R2 para armazenamento dos arquivos. A validação técnica foi realizada por meio de testes automatizados no backend e no frontend, complementados por testes de integração com PostgreSQL, migrações e fluxos HTTP. Foram executados e aprovados 418 testes distintos no backend e 159 no frontend, totalizando 577 casos de teste automatizados. A usabilidade das interfaces também foi analisada por meio de inspeção heurística baseada nas dez heurísticas de Nielsen, utilizando evidências visuais das principais funcionalidades da plataforma. Os resultados demonstraram o funcionamento dos comportamentos contemplados pelas suítes de testes e a aderência das interfaces analisadas aos princípios heurísticos adotados. Conclui-se que o RepolF constitui uma solução funcional para o escopo estabelecido, reunindo recursos para organização, gerenciamento, compartilhamento e recuperação de materiais didáticos em um ambiente estruturado.

**Palavras-chave:** materiais didáticos digitais; repositórios digitais; plataformas web; compartilhamento de conhecimento; usabilidade.

## ABSTRACT

The dispersion of teaching materials across different digital channels can make their organization, location, and retrieval difficult in academic environments. In this context, this study aimed to develop and technically validate RepolF, a web platform designed for the centralized management and sharing of teaching materials at the Federal Institute of Piauí (IFPI) - Picos Campus. The solution was developed to support the management of subjects, materials, classes, and assignments, as well as authentication, authorization, access control, administration, and moderation mechanisms. The research is characterized as applied, with a qualitative approach and exploratory and descriptive objectives. The development was conducted incrementally, involving requirements elicitation and specification, modeling, architecture definition, implementation, and validation. The solution uses React for the frontend, NestJS for the backend, PostgreSQL for structured data persistence, and Cloudflare R2 for file storage. Technical validation was performed through automated tests on the backend and frontend, complemented by integration tests with PostgreSQL, database migrations, and HTTP workflows. A total of 418 distinct backend tests and 159 frontend tests were executed and approved, resulting in 577 automated test cases. The usability of the interfaces was also analyzed through a heuristic inspection based on Nielsen's ten usability heuristics, using visual evidence from the platform's main functionalities. The results demonstrated the proper functioning of the behaviors covered by the test suites and the adherence of the analyzed interfaces to the adopted heuristic principles. It is concluded that RepolF constitutes a functional solution within the established scope, providing resources for the organization, management, sharing, and retrieval of teaching materials in a structured environment.

**Keywords:** digital teaching materials; digital repositories; web platforms; knowledge sharing; usability.

## LISTA DE ILUSTRAÇÕES

|           |                                                                           |    |
|-----------|---------------------------------------------------------------------------|----|
| Figura 1  | – Casos de uso de acesso, conta e funcionalidades comuns do RepolF.....   | 54 |
| Figura 2  | – Casos de uso do estudante no RepolF.....                                | 55 |
| Figura 3  | – Casos de uso do professor no RepolF.....                                | 56 |
| Figura 4  | – Casos de uso do administrador no RepolF.....                            | 57 |
| Figura 5  | – Modelo Entidade-Relacionamento do RepolF.....                           | 58 |
| Figura 6  | – Arquitetura geral do RepolF.....                                        | 63 |
| Figura 7  | – Fluxo de upload de arquivos no RepolF.....                              | 66 |
| Figura 8  | – Interfaces de autenticação e gerenciamento de conta do RepolF.....      | 71 |
| Figura 9  | – Perfil de usuário e consulta de professores no RepolF.....              | 72 |
| Figura 10 | – Painel de disciplinas do professor.....                                 | 73 |
| Figura 11 | – Gerenciamento de materiais de uma disciplina.....                       | 74 |
| Figura 12 | – Gerenciamento de turma no RepolF.....                                   | 75 |
| Figura 13 | – Gerenciamento de atividades no RepolF.....                              | 76 |
| Figura 14 | – Detalhes de atividade e acompanhamento de entregas.....                 | 77 |
| Figura 15 | – Interface de moderação de denúncias.....                                | 78 |
| Figura 16 | – Painel administrativo do RepolF.....                                    | 79 |
| Figura 17 | – <i>Feedback</i> apresentado ao usuário após uma operação no RepolF..... | 87 |
| Figura 18 | – Utilização de terminologia acadêmica nas interfaces do RepolF.          | 88 |
| Figura 19 | – Controle do usuário em operação que exige confirmação.....              | 89 |
| Figura 20 | – Padronização dos elementos de interface do RepolF.....                  | 90 |
| Figura 21 | – Mecanismo de prevenção de erros em uma operação do                      | 91 |

|           |                                                                          |    |
|-----------|--------------------------------------------------------------------------|----|
|           | RepolF.....                                                              |    |
| Figura 22 | - Apresentação contextual de informações e ações no RepolF.....          | 92 |
| Figura 23 | - Recursos de pesquisa e filtragem disponíveis no RepolF.....            | 93 |
| Figura 24 | - Organização visual de uma das principais interfaces do RepolF.         | 94 |
| Figura 25 | - Tratamento de situação de erro apresentado pelo RepolF.....            | 95 |
| Figura 26 | - Orientação contextual apresentada em uma funcionalidade do RepolF..... | 96 |

## LISTA DE QUADROS

|           |   |                                                                  |    |
|-----------|---|------------------------------------------------------------------|----|
| Quadro 1  | – | Atores do RepolF.....                                            | 44 |
| Quadro 2  | – | Requisitos funcionais de autenticação e perfil.....              | 45 |
| Quadro 3  | – | Requisitos funcionais de disciplinas e materiais.....            | 45 |
| Quadro 4  | – | Requisitos funcionais de turmas.....                             | 46 |
| Quadro 5  | – | Requisitos funcionais de atividades e entregas.....              | 47 |
| Quadro 6  | – | Requisitos funcionais de administração e moderação.....          | 47 |
| Quadro 7  | – | Requisitos não funcionais.....                                   | 48 |
| Quadro 8  | – | Regras de negócio de contas e autenticação.....                  | 50 |
| Quadro 9  | – | Regras de negócio de disciplinas e materiais.....                | 50 |
| Quadro 10 | – | Regras de negócio de turmas.....                                 | 51 |
| Quadro 11 | – | Regras de negócio de atividades e entregas.....                  | 52 |
| Quadro 12 | – | Regras de negócio de administração e denúncias.....              | 53 |
| Quadro 13 | – | Entidades persistidas pelo RepolF.....                           | 59 |
| Quadro 14 | – | Principais tecnologias utilizadas no RepolF.....                 | 68 |
| Quadro 15 | – | Síntese dos testes automatizados do <i>backend</i> .....         | 82 |
| Quadro 16 | – | Síntese dos testes automatizados do <i>frontend</i> .....        | 83 |
| Quadro 17 | – | Síntese da validação automatizada do RepolF.....                 | 84 |
| Quadro 18 | – | Resultado dos testes com PostgreSQL.....                         | 85 |
| Quadro 19 | – | Relação entre os objetivos específicos e os resultados obtidos.. | 97 |

## LISTA DE ABREVIATURAS E SIGLAS

|      |                                                              |
|------|--------------------------------------------------------------|
| API  | Application Programming Interface                            |
| CORS | Cross-Origin Resource Sharing                                |
| CSS  | Cascading Style Sheets                                       |
| DOM  | Document Object Model                                        |
| GED  | Gestão Eletrônica de Documentos                              |
| HTTP | Hypertext Transfer Protocol                                  |
| IFPI | Instituto Federal de Educação, Ciência e Tecnologia do Piauí |
| JSON | JavaScript Object Notation                                   |
| JWT  | JSON Web Token                                               |
| MDD  | Materiais Didáticos Digitais                                 |
| ORM  | Object-Relational Mapping                                    |
| QR   | Quick Response                                               |
| REST | Representational State Transfer                              |
| RF   | Requisito Funcional                                          |
| RN   | Regra de Negócio                                             |
| RNF  | Requisito Não Funcional                                      |
| TDIC | Tecnologias Digitais de Informação e Comunicação             |
| URL  | Uniform Resource Locator                                     |
| UUID | Universally Unique Identifier                                |

## SUMÁRIO

|                                                                                   |           |
|-----------------------------------------------------------------------------------|-----------|
| <b>1 INTRODUÇÃO.....</b>                                                          | <b>15</b> |
| <b>2 OBJETIVOS.....</b>                                                           | <b>18</b> |
| 2.1 OBJETIVO GERAL.....                                                           | 18        |
| 2.2 OBJETIVOS ESPECÍFICOS.....                                                    | 18        |
| <b>3 FUNDAMENTAÇÃO TEÓRICA.....</b>                                               | <b>20</b> |
| 3.1 TECNOLOGIAS DIGITAIS E GESTÃO DA INFORMAÇÃO NA EDUCAÇÃO.....                  | 20        |
| 3.2 MATERIAIS DIDÁTICOS DIGITAIS COMO INSTRUMENTOS DE MEDIAÇÃO.....               | 21        |
| 3.3 PLATAFORMAS DIGITAIS E COMPARTILHAMENTO DE INFORMAÇÕES E<br>CONHECIMENTO..... | 22        |
| 3.4 GESTÃO ELETRÔNICA DE DOCUMENTOS NO AMBIENTE EDUCACIONAL.....                  | 24        |
| 3.5 REPOSITÓRIOS DIGITAIS E ORGANIZAÇÃO DA INFORMAÇÃO.....                        | 25        |
| 3.6 SISTEMAS DE INFORMAÇÃO E PLATAFORMAS WEB.....                                 | 27        |
| 3.7 ENGENHARIA DE SOFTWARE E ARQUITETURA DE SISTEMAS WEB.....                     | 28        |
| 3.8 USABILIDADE E AVALIAÇÃO HEURÍSTICA.....                                       | 30        |
| <b>4 METODOLOGIA.....</b>                                                         | <b>33</b> |
| 4.1 CLASSIFICAÇÃO DA PESQUISA.....                                                | 33        |
| 4.2 CONTEXTO DA PESQUISA.....                                                     | 34        |
| 4.3 PROCESSO DE DESENVOLVIMENTO DO SOFTWARE.....                                  | 34        |
| 4.4 FONTES DE EVIDÊNCIA E DOCUMENTAÇÃO DO SISTEMA.....                            | 35        |
| 4.5 VALIDAÇÃO TÉCNICA POR TESTES AUTOMATIZADOS.....                               | 36        |
| 4.5.1 Testes automatizados do backend.....                                        | 36        |
| 4.5.2 Testes automatizados do frontend.....                                       | 37        |
| 4.5.3 Testes de integração com PostgreSQL.....                                    | 39        |
| 4.6 AVALIAÇÃO DE USABILIDADE POR INSPEÇÃO HEURÍSTICA.....                         | 40        |
| 4.7 PARTICIPANTES E ASPECTOS ÉTICOS.....                                          | 41        |
| <b>5 REQUISITOS E MODELAGEM DO SISTEMA.....</b>                                   | <b>43</b> |
| 5.1 ESCOPO DO SISTEMA.....                                                        | 43        |
| 5.2 ATORES DO SISTEMA.....                                                        | 44        |
| 5.3 REQUISITOS FUNCIONAIS.....                                                    | 44        |
| 5.4 REQUISITOS NÃO FUNCIONAIS.....                                                | 48        |
| 5.5 REGRAS DE NEGÓCIO.....                                                        | 49        |
| 5.5.1 Contas e autenticação.....                                                  | 50        |
| 5.5.2 Disciplinas e materiais.....                                                | 50        |
| 5.5.3 Turmas.....                                                                 | 51        |
| 5.5.4 Atividades e entregas.....                                                  | 52        |
| 5.5.5 Administração e denúncias.....                                              | 52        |
| 5.6 MODELAGEM DE CASOS DE USO.....                                                | 53        |
| 5.6.1 Acesso, conta e funcionalidades comuns.....                                 | 53        |
| 5.6.2 Casos de uso do estudante.....                                              | 54        |
| 5.6.3 Casos de uso do professor.....                                              | 55        |
| 5.6.4 Casos de uso do administrador.....                                          | 56        |
| 5.7 MODELAGEM DOS DADOS.....                                                      | 57        |

|                                                               |            |
|---------------------------------------------------------------|------------|
| 5.7.1 Entidades do sistema.....                               | 59         |
| 5.7.2 Relacionamentos e cardinalidades.....                   | 60         |
| 5.7.3 Integridade e restrições do modelo.....                 | 61         |
| <b>6 ARQUITETURA E TECNOLOGIAS.....</b>                       | <b>63</b>  |
| 6.1 ARQUITETURA GERAL DA APLICAÇÃO.....                       | 63         |
| 6.2 FRONTEND E BACKEND.....                                   | 64         |
| 6.3 PERSISTÊNCIA E ARMAZENAMENTO DE ARQUIVOS.....             | 65         |
| 6.4 SEGURANÇA E SERVIÇOS EXTERNOS.....                        | 67         |
| 6.5 TECNOLOGIAS UTILIZADAS.....                               | 68         |
| <b>7 IMPLEMENTAÇÃO E FUNCIONALIDADES DO REPOIF.....</b>       | <b>70</b>  |
| 7.1 AUTENTICAÇÃO E GERENCIAMENTO DE CONTAS.....               | 70         |
| 7.2 PERFIS E CONSULTA DE PROFESSORES.....                     | 71         |
| 7.3 DISCIPLINAS E GERENCIAMENTO DE MATERIAIS.....             | 72         |
| 7.4 TURMAS, DISCIPLINAS E INGRESSO DE ESTUDANTES.....         | 74         |
| 7.5 ATIVIDADES E ENTREGAS.....                                | 76         |
| 7.6 DENÚNCIAS E MODERAÇÃO.....                                | 77         |
| 7.7 FUNCIONALIDADES ADMINISTRATIVAS.....                      | 78         |
| 7.8 SÍNTESE DAS FUNCIONALIDADES IMPLEMENTADAS.....            | 80         |
| <b>8 RESULTADOS E VALIDAÇÃO.....</b>                          | <b>81</b>  |
| 8.1 RESULTADOS DA IMPLEMENTAÇÃO.....                          | 81         |
| 8.2 VALIDAÇÃO TÉCNICA.....                                    | 81         |
| 8.2.1 Resultados dos testes do backend.....                   | 82         |
| 8.2.2 Resultados dos testes do frontend.....                  | 83         |
| 8.2.3 Síntese dos testes automatizados.....                   | 84         |
| 8.3 TESTES DE INTEGRAÇÃO.....                                 | 85         |
| 8.4 AVALIAÇÃO HEURÍSTICA DE USABILIDADE.....                  | 86         |
| 8.4.1 Visibilidade do estado do sistema.....                  | 86         |
| 8.4.2 Correspondência entre o sistema e o mundo real.....     | 87         |
| 8.4.3 Controle e liberdade do usuário.....                    | 88         |
| 8.4.4 Consistência e padrões.....                             | 89         |
| 8.4.5 Prevenção de erros.....                                 | 90         |
| 8.4.6 Reconhecimento em vez de memorização.....               | 91         |
| 8.4.7 Flexibilidade e eficiência de uso.....                  | 92         |
| 8.4.8 Design estético e minimalista.....                      | 93         |
| 8.4.9 Reconhecimento, diagnóstico e recuperação de erros..... | 94         |
| 8.4.10 Ajuda e documentação.....                              | 95         |
| 8.4.11 Síntese da avaliação heurística.....                   | 96         |
| 8.5 ATENDIMENTO AOS OBJETIVOS.....                            | 97         |
| <b>9 CONSIDERAÇÕES FINAIS.....</b>                            | <b>99</b>  |
| 9.1 TRABALHOS FUTUROS.....                                    | 101        |
| <b>REFERÊNCIAS.....</b>                                       | <b>103</b> |

## 1 INTRODUÇÃO

A presença das tecnologias digitais transformou os processos de ensino e aprendizagem, influenciando a produção, o acesso e o compartilhamento do conhecimento. Nesse contexto, esses recursos passaram de complementares a integrantes das práticas pedagógicas e da organização da informação. Gomes (2025) destaca que seu uso planejado contribui para práticas mais alinhadas às necessidades da educação contemporânea.

Os materiais didáticos digitais assumem importância por reunirem conteúdos utilizados por professores e estudantes, como textos, apresentações, vídeos e outros documentos. Colusso e Astudillo (2023) compreendem esses materiais como instrumentos de mediação entre docentes e discentes, auxiliando na continuidade dos estudos e no acesso aos conteúdos. Assim, sua utilização envolve não apenas o armazenamento, mas também sua organização, disponibilização e recuperação.

Para utilizar adequadamente os materiais didáticos digitais, são necessários ambientes estruturados para sua classificação, armazenamento e localização. Segundo Santos (2023), as plataformas digitais ampliam a flexibilidade de acesso aos conteúdos, permitindo que os estudantes os consultem conforme sua disponibilidade e ritmo de estudo. Os repositórios digitais também contribuem ao reunir conteúdos e metadados, oferecendo mecanismos de pesquisa, recuperação e controle de acesso (Rodrigues et al., 2011).

Apesar das possibilidades das tecnologias digitais, o uso de diferentes canais para compartilhar materiais acadêmicos pode provocar sua dispersão e dificultar a localização dos arquivos. Essa fragmentação pode resultar em perda, duplicação de documentos e retrabalho. Alves et al. (2022) ressaltam a importância de ferramentas e processos adequados para o compartilhamento do conhecimento, enquanto Santos et al. (2022) destacam que o acúmulo desordenado de documentos compromete sua organização e recuperação.

No contexto deste trabalho, o compartilhamento de materiais didáticos entre professores e estudantes do IFPI - Campus Picos pode ocorrer por diferentes meios digitais, sem que os conteúdos estejam reunidos em um ambiente específico. Essa

dispersão dificulta o acesso contínuo aos materiais, especialmente quando envolvem diferentes disciplinas, turmas e períodos acadêmicos. Diante dessa problemática, estabeleceu-se a seguinte questão de pesquisa: Como uma plataforma *web* pode contribuir para a centralização, a organização e a recuperação estruturada de materiais didáticos no IFPI - Campus Picos?

Diante desse cenário, este trabalho teve como objetivo desenvolver o RepolF, uma aplicação *web* voltada ao gerenciamento e compartilhamento centralizado de materiais didáticos, com recursos de armazenamento, organização, recuperação e controle de acesso.

Como resultado, foram implementadas funcionalidades para o gerenciamento de disciplinas, turmas, materiais e atividades, além de mecanismos de autenticação, autorização e recursos administrativos e de moderação.

A relevância do trabalho está na possibilidade de reunir, em um mesmo ambiente, recursos que normalmente permanecem distribuídos entre diferentes ferramentas. No âmbito educacional, o RepolF organiza materiais por disciplinas, turmas e atividades, relacionando os conteúdos ao contexto em que são utilizados. No âmbito tecnológico, seu desenvolvimento possibilitou a aplicação de conceitos de engenharia de *software*, arquitetura *web*, modelagem de dados, segurança, armazenamento de arquivos, testes automatizados e avaliação de usabilidade.

Quanto aos procedimentos metodológicos, a pesquisa caracteriza-se como aplicada, com abordagem qualitativa e objetivos exploratórios e descritivos. O desenvolvimento do *software* ocorreu de forma incremental, envolvendo levantamento e refinamento de requisitos, modelagem, implementação e avaliação.

A validação foi realizada por testes automatizados no *backend* e *frontend*, verificando funcionalidades, regras de negócio e mecanismos de segurança, além de avaliação heurística de usabilidade baseada nos princípios de Nielsen (1994). Essas estratégias permitiram analisar o funcionamento técnico e a interação com a interface, mantendo o foco do RepolF na gestão e no compartilhamento estruturado de materiais didáticos.

Além desta Introdução, o trabalho está organizado em outros oito capítulos. O

Capítulo 2 apresenta o objetivo geral e os objetivos específicos. O Capítulo 3 reúne a fundamentação teórica sobre tecnologias digitais na educação, materiais didáticos digitais, compartilhamento do conhecimento, gestão eletrônica de documentos, repositórios digitais, sistemas de informação e engenharia de *software*. O Capítulo 4 descreve a metodologia empregada na pesquisa e no desenvolvimento. O Capítulo 5 apresenta os requisitos, as regras de negócio e a modelagem do sistema. O Capítulo 6 detalha a arquitetura e as tecnologias utilizadas. O Capítulo 7 apresenta a implementação do RepolF e descreve suas principais funcionalidades. O Capítulo 8 apresenta os resultados obtidos e os procedimentos de validação da aplicação. Por fim, o Capítulo 9 reúne as considerações finais, as limitações identificadas e as possibilidades de trabalhos futuros.

## 2 OBJETIVOS

Os objetivos deste trabalho foram definidos a partir do problema relacionado à fragmentação do compartilhamento de materiais didáticos em diferentes meios digitais e considerando o escopo alcançado durante o desenvolvimento do RepolF. Além da centralização e organização dos conteúdos acadêmicos, o sistema incorporou funcionalidades relacionadas ao gerenciamento de turmas e atividades, controle de acesso, administração e moderação da plataforma, bem como mecanismos de validação técnica e avaliação de usabilidade.

### 2.1 OBJETIVO GERAL

Desenvolver e validar tecnicamente o RepolF, uma plataforma *web* voltada ao gerenciamento e ao compartilhamento centralizado de materiais didáticos no Instituto Federal de Educação, Ciência e Tecnologia do Piauí - Campus Picos, contemplando funcionalidades para organização de conteúdos acadêmicos, gerenciamento de disciplinas e turmas, atividades, controle de acesso, administração e moderação da plataforma.

### 2.2 OBJETIVOS ESPECÍFICOS

- Realizar levantamento bibliográfico sobre tecnologias digitais aplicadas à educação, materiais didáticos digitais, compartilhamento de conhecimento, gestão eletrônica de documentos, repositórios digitais, engenharia de *software* e usabilidade;
- Levantar, analisar e especificar os requisitos funcionais e não funcionais necessários à construção de uma plataforma *web* destinada à organização, ao armazenamento, ao compartilhamento e à recuperação de materiais didáticos;
- Definir a arquitetura da aplicação, a modelagem dos dados e a organização dos componentes necessários à estruturação da plataforma, considerando a separação entre dados estruturados e arquivos digitais;
- Estruturar mecanismos de autenticação, autorização e controle de acesso

compatíveis com os diferentes perfis de usuários da plataforma;

- Estruturar o gerenciamento de disciplinas e materiais acadêmicos, contemplando sua organização, disponibilização, busca, recuperação e controle de acesso;
- Estruturar o gerenciamento de turmas e sua relação com disciplinas e estudantes, considerando mecanismos de ingresso e controle de membros;
- Estruturar o gerenciamento de atividades acadêmicas, contemplando definição de prazos, disponibilização de anexos, recebimento de entregas e controle de reenvios;
- Estruturar recursos de administração e moderação voltados ao gerenciamento de usuários e arquivos, à manutenção dos registros e ao tratamento de denúncias;
- Incorporar mecanismos de segurança relacionados à autenticação, autorização, validação de arquivos, controle de requisições e proteção dos recursos da plataforma;
- Realizar testes de *software* no *backend* e no *frontend* para verificar o funcionamento das funcionalidades, regras de negócio e mecanismos de segurança implementados;
- Avaliar a usabilidade das principais interfaces e fluxos de interação do RepolF por meio de inspeção heurística baseada nas heurísticas de Nielsen;
- Analisar o atendimento aos requisitos e objetivos estabelecidos para a solução a partir dos resultados obtidos nos testes de software e na avaliação de usabilidade.

### 3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos que fundamentam a problemática investigada e o desenvolvimento do RepolIF. São discutidos aspectos relacionados às tecnologias digitais no contexto educacional, aos materiais didáticos digitais, ao compartilhamento de informações e conhecimento, à Gestão Eletrônica de Documentos, aos repositórios digitais, aos sistemas de informação, à Engenharia de Software e à usabilidade de sistemas interativos. Esses conceitos fornecem a base teórica necessária para compreender tanto o problema relacionado à dispersão dos materiais acadêmicos quanto as decisões envolvidas na construção e na avaliação da plataforma.

#### 3.1 TECNOLOGIAS DIGITAIS E GESTÃO DA INFORMAÇÃO NA EDUCAÇÃO

A presença das tecnologias digitais em diferentes setores da sociedade modificou as formas de produzir, armazenar, acessar e compartilhar informações. No contexto educacional, essas tecnologias passaram a integrar atividades desenvolvidas por professores e estudantes, possibilitando novas formas de disponibilização de conteúdos e de interação com recursos utilizados durante o processo de ensino e aprendizagem.

Gomes (2025) destaca que a incorporação das tecnologias digitais ao contexto escolar pode contribuir para a construção de práticas mais alinhadas às demandas contemporâneas. Segundo o autor, esses recursos podem ser empregados na produção e no aperfeiçoamento de materiais pedagógicos, ampliando as possibilidades de utilização de diferentes formatos e linguagens. Entretanto, sua aplicação deve estar associada a objetivos educacionais definidos, de modo que a tecnologia não seja utilizada apenas como um recurso isolado.

O avanço das Tecnologias Digitais de Informação e Comunicação (TDIC) também ampliou as possibilidades de acesso e distribuição dos conteúdos educacionais. Santos (2023) ressalta que as plataformas digitais proporcionam maior flexibilidade no acesso às informações, permitindo que estudantes consultem materiais e realizem atividades em diferentes horários e espaços. Dessa forma, os recursos digitais contribuem para reduzir a dependência do ambiente físico como

único local de acesso aos conteúdos acadêmicos.

Ao mesmo tempo em que as tecnologias facilitam a produção e a circulação das informações, o crescimento da quantidade de conteúdos digitais torna necessária a adoção de mecanismos que permitam organizá-los adequadamente. Em uma instituição de ensino, diferentes materiais podem ser produzidos e compartilhados ao longo de disciplinas, turmas e períodos acadêmicos. Quando esses conteúdos não seguem uma estrutura de organização, sua localização posterior pode se tornar mais difícil.

Nesse contexto, a gestão da informação assume importância por envolver processos relacionados à organização, ao armazenamento e à recuperação dos conteúdos produzidos. A utilização de sistemas computacionais permite estruturar essas informações e estabelecer relações entre os diferentes recursos, oferecendo mecanismos para sua organização e recuperação.

Portanto, o uso das tecnologias digitais no ambiente educacional não está relacionado apenas à criação ou disponibilização dos materiais. Também envolve a forma como as informações são organizadas e mantidas acessíveis ao longo do tempo. Essa necessidade torna-se especialmente relevante quando se considera a quantidade e a diversidade de materiais didáticos utilizados em ambientes acadêmicos.

### 3.2 MATERIAIS DIDÁTICOS DIGITAIS COMO INSTRUMENTOS DE MEDIAÇÃO

A evolução dos recursos tecnológicos ampliou o conceito de material didático para além dos suportes impressos. Documentos textuais, apresentações, imagens, vídeos, arquivos de áudio, códigos-fonte e outros recursos digitais passaram a ser utilizados como instrumentos de apoio às atividades acadêmicas.

Colusso e Astudillo (2023) abordam os Materiais Didáticos Digitais (MDD) como recursos capazes de atuar na mediação das interações entre professores e estudantes. Os autores analisaram a utilização de diferentes tipos de arquivos no ambiente virtual Moodle da Universidade Federal de Santa Maria e verificaram a presença significativa de materiais digitais nas interações realizadas entre docentes e discentes.

Os materiais digitais podem ser disponibilizados por diferentes meios, como ambientes virtuais de aprendizagem, repositórios, redes de colaboração, correio eletrônico e redes sociais. Essa característica amplia as possibilidades de acesso e reutilização dos conteúdos, permitindo que um mesmo recurso seja empregado em diferentes momentos e contextos educacionais.

Gomes (2025) também destaca que a utilização das tecnologias digitais pode favorecer a produção de materiais mais contextualizados e permitir maior autonomia ao professor na elaboração dos recursos utilizados em sua prática. A possibilidade de produzir e adaptar conteúdos digitais permite que diferentes formatos sejam empregados conforme as necessidades relacionadas ao conteúdo e ao contexto em que serão utilizados.

Entretanto, o valor de um material didático digital não está relacionado apenas ao arquivo armazenado. O contexto associado ao recurso também possui importância. Um documento utilizado em determinada disciplina, por exemplo, pode possuir informações relacionadas ao professor responsável, ao conteúdo abordado, à turma para a qual foi disponibilizado e ao período em que foi utilizado.

Dessa maneira, a organização dos materiais digitais é necessária para preservar essas relações e facilitar sua localização posterior. A simples existência de arquivos em formato digital não garante que eles sejam facilmente recuperados ou reutilizados. Para isso, torna-se necessário associá-los a informações que permitam identificar seu conteúdo e o contexto acadêmico ao qual pertencem.

Essa característica aproxima os materiais didáticos digitais dos conceitos de gestão da informação e de repositórios digitais, uma vez que o armazenamento estruturado e a utilização de informações descritivas podem contribuir para que esses recursos permaneçam acessíveis e identificáveis.

### 3.3 PLATAFORMAS DIGITAIS E COMPARTILHAMENTO DE INFORMAÇÕES E CONHECIMENTO

As plataformas digitais constituem ambientes tecnológicos que possibilitam interação, disponibilização de conteúdos e troca de informações entre seus usuários. No contexto educacional, essas plataformas podem assumir diferentes funções,

incluindo acesso a materiais, realização de atividades e apoio à comunicação entre os participantes.

Santos (2023) aponta que as plataformas digitais podem proporcionar ambientes mais flexíveis para acesso aos conteúdos educacionais. A possibilidade de consultar informações por meio da Internet permite que os materiais permaneçam disponíveis além do momento em que foram inicialmente apresentados ou compartilhados, contribuindo para ampliar as possibilidades de acesso dos estudantes.

O compartilhamento de informações possui papel relevante no ambiente escolar, no qual professores, estudantes e demais participantes necessitam trocar diferentes tipos de conteúdos. Alves *et al.* (2022) destacam que o ambiente educacional demanda intensa circulação de informações e conhecimento, sendo necessária a existência de ferramentas, processos e rotinas adequadas que contribuam para esse compartilhamento.

Em estudo realizado durante o período de ensino remoto emergencial, Alves *et al.* (2022) observaram a utilização de diferentes tecnologias digitais para manter a comunicação e o compartilhamento de conteúdos entre professores, estudantes e responsáveis. Embora o estudo esteja relacionado ao contexto específico da pandemia de Covid-19, ele evidencia a importância adquirida pelas tecnologias digitais para a distribuição de informações e materiais utilizados nas atividades educacionais.

Entretanto, a disponibilidade de diferentes ferramentas também pode resultar na fragmentação das informações. Um mesmo material pode ser compartilhado por aplicativo de mensagens, armazenado em um serviço de nuvem e posteriormente enviado novamente por correio eletrônico. Apesar de esses recursos permitirem a transmissão dos arquivos, não necessariamente oferecem uma estrutura comum para classificação e recuperação dos conteúdos.

Nesse sentido, centralizar informações não significa apenas reunir arquivos em um mesmo local de armazenamento. Também envolve estabelecer relações entre os conteúdos e o contexto no qual são utilizados. A associação dos materiais às disciplinas, aos responsáveis pela publicação e aos usuários que possuem

acesso contribui para criar uma estrutura mais organizada para a recuperação das informações.

Assim, plataformas destinadas especificamente ao gerenciamento de materiais acadêmicos podem atuar como mecanismos de organização e compartilhamento. Essa abordagem aproxima-se dos princípios da Gestão Eletrônica de Documentos, que considera não apenas o armazenamento dos arquivos, mas também os processos necessários para classificá-los, localizá-los e disponibilizá-los.

### 3.4 GESTÃO ELETRÔNICA DE DOCUMENTOS NO AMBIENTE EDUCACIONAL

O aumento da quantidade de documentos produzidos digitalmente exige mecanismos que permitam organizar, armazenar e recuperar essas informações. Nesse contexto, a Gestão Eletrônica de Documentos (GED) reúne tecnologias e procedimentos utilizados para o gerenciamento de documentos em ambiente digital.

Santos *et al.* (2022) apresentam a GED como um sistema que possibilita a captação, o armazenamento, a localização e o gerenciamento de versões digitais das informações. A utilização dessas tecnologias busca reduzir dificuldades relacionadas à organização documental e facilitar a recuperação dos conteúdos armazenados.

Um dos problemas associados à ausência de uma gestão documental adequada é o acúmulo desordenado de documentos. Santos *et al.* (2022) apontam que essa situação pode comprometer a agilidade na localização das informações, dificultando seu uso quando necessário. No ambiente digital, problema semelhante pode ocorrer quando grandes quantidades de arquivos são armazenadas sem uma estrutura que permita identificá-los e classificá-los adequadamente.

No contexto educacional, diferentes tipos de documentos são produzidos e utilizados durante as atividades acadêmicas. Apresentações, exercícios, documentos textuais, códigos, orientações, materiais complementares e outros arquivos podem acumular-se ao longo do tempo. A organização desses conteúdos precisa permitir que professores e estudantes identifiquem a qual contexto cada material pertence.

Outro aspecto relevante da Gestão Eletrônica de Documentos refere-se ao controle de acesso às informações. Determinados documentos podem estar destinados a grupos específicos de usuários, enquanto outros podem possuir maior nível de visibilidade. Dessa forma, mecanismos de autenticação, autorização e controle de acesso podem fazer parte de uma estratégia de gerenciamento das informações armazenadas.

Santos *et al.* (2022) destacam ainda que a adoção de tecnologias de gestão documental pode facilitar o acesso às informações e sua recuperação, além de contribuir para reduzir dificuldades decorrentes da organização inadequada dos documentos.

Dessa maneira, a GED fornece princípios importantes para sistemas destinados ao gerenciamento de materiais didáticos. Organizar documentos de forma estruturada, associá-los a informações descritivas e estabelecer mecanismos de acesso são elementos que contribuem para transformar um conjunto de arquivos em um ambiente organizado de informações.

### 3.5 REPOSITÓRIOS DIGITAIS E ORGANIZAÇÃO DA INFORMAÇÃO

Os repositórios digitais podem ser compreendidos como uma aplicação específica dos sistemas de informação voltada ao gerenciamento de recursos digitais. Enquanto os sistemas de informação organizam e processam dados em diferentes contextos organizacionais (Laudon; Laudon, 2023), os repositórios concentram-se na preservação, organização e recuperação de objetos digitais e de seus respectivos metadados (Rodrigues; Taga; Vieira, 2011).

Rodrigues, Taga e Vieira (2011) apresentam os repositórios digitais como sistemas de informação capazes de armazenar e disponibilizar recursos digitais. Os autores destacam que um repositório oferece mecanismos relacionados à identificação, ao armazenamento e à recuperação dos conteúdos, diferenciando-se de uma simples coleção de arquivos.

Entre as características dos repositórios discutidas pelos autores encontram-se o depósito de conteúdos, o gerenciamento dos próprios recursos e de seus metadados, além da disponibilização de serviços como inserção, localização,

pesquisa e controle de acesso. Essas características demonstram que a organização das informações possui papel central na construção de um repositório digital.

Nesse contexto, os metadados desempenham uma função importante. De maneira geral, correspondem a dados utilizados para identificar, descrever e contextualizar outros recursos. Em ambientes destinados ao armazenamento de materiais acadêmicos, informações como nome do arquivo, autoria, formato, disciplina relacionada e responsável pela publicação podem auxiliar na identificação e recuperação dos conteúdos.

Rodrigues, Taga e Vieira (2011) destacam que os metadados podem ser utilizados para identificar, descrever, localizar, integrar e organizar recursos digitais. Os autores também ressaltam a importância das relações entre objetos de aprendizagem, disciplinas, cursos, autores e instituições para que seja possível contextualizar os conteúdos armazenados. A estruturação dessas informações amplia as possibilidades de recuperação e reutilização dos materiais.

A utilização dos metadados é particularmente relevante quando a quantidade de arquivos armazenados aumenta. A existência de um grande conjunto de documentos sem informações descritivas pode reproduzir, no meio digital, os mesmos problemas associados ao arquivamento desorganizado. Por outro lado, a associação de informações estruturadas aos arquivos permite utilizar mecanismos de busca, classificação e filtragem.

Os repositórios educacionais também podem apresentar diferentes formas de controle sobre a disponibilidade de seus conteúdos. Embora Rodrigues, Taga e Vieira (2011) concentrem sua análise nos Repositórios Educacionais Abertos, os princípios relacionados à organização dos recursos, metadados, pesquisa e recuperação podem ser utilizados em sistemas que estabeleçam diferentes níveis de acesso aos seus conteúdos.

Os autores ainda observam que apenas a adoção de tecnologias não é suficiente para solucionar questões relacionadas à comunicação, organização e representação da informação, sendo necessária uma estrutura adequada para o gerenciamento dos conteúdos digitais.

Portanto, os conceitos relacionados aos repositórios digitais oferecem uma base relevante para sistemas destinados à centralização de materiais didáticos. A associação entre conteúdo, informações descritivas e mecanismos de recuperação possibilita que os arquivos sejam organizados de forma mais consistente e permaneçam acessíveis dentro do contexto ao qual pertencem.

### 3.6 SISTEMAS DE INFORMAÇÃO E PLATAFORMAS WEB

Os sistemas de informação podem ser compreendidos como estruturas que integram recursos destinados à coleta, ao processamento, ao armazenamento e à distribuição de informações (Laudon; Laudon, 2023). Esses sistemas são utilizados em diferentes tipos de organizações para estruturar dados e apoiar a execução de seus processos.

Laudon e Laudon (2023) destacam que os sistemas de informação exercem papel importante na organização e no fluxo dos dados em ambientes institucionais. Ao permitir que as informações sejam armazenadas e processadas de maneira estruturada, esses sistemas contribuem para sua recuperação e utilização de acordo com as necessidades organizacionais.

No contexto educacional, os sistemas de informação podem apoiar diferentes atividades. Quando aplicados ao gerenciamento de conteúdos acadêmicos, permitem representar elementos como usuários, disciplinas, materiais, turmas e atividades, além das relações existentes entre esses elementos (Laudon; Laudon, 2023).

Essa característica diferencia um sistema de informação de um mecanismo destinado apenas ao armazenamento de arquivos. Enquanto o armazenamento está relacionado principalmente à persistência do conteúdo, um sistema de informação estabelece estruturas e regras que determinam como os dados são cadastrados, relacionados, consultados e apresentados aos usuários (Laudon; Laudon, 2023).

Os sistemas *web* representam uma das formas de disponibilização dessas soluções. Por meio de navegadores, os usuários podem acessar as funcionalidades oferecidas por uma aplicação sem que seja necessária a instalação de um programa específico em cada dispositivo. A comunicação realizada por meio da *Internet*

permite que diferentes componentes da solução interajam para processar as solicitações e recuperar as informações necessárias (Sommerville, 2011).

Em aplicações *web*, é comum estabelecer uma separação entre a interface apresentada ao usuário, os componentes responsáveis pela aplicação das regras de negócio e os mecanismos destinados à persistência dos dados. Essa organização possibilita distribuir responsabilidades entre diferentes partes do sistema e facilita sua manutenção e evolução (Sommerville, 2011).

No contexto do gerenciamento de materiais acadêmicos, uma plataforma *web* pode atuar como o ponto de interação entre os usuários e os diferentes recursos armazenados. A partir dessa interface, os conteúdos e suas informações associadas podem ser apresentados de maneira estruturada, respeitando as relações e permissões estabelecidas pelo sistema (Laudon; Laudon, 2023; Sommerville, 2011).

### 3.7 ENGENHARIA DE SOFTWARE E ARQUITETURA DE SISTEMAS WEB

O desenvolvimento de uma aplicação computacional envolve um conjunto de atividades destinadas a transformar necessidades identificadas em um sistema funcional. A Engenharia de *Software* reúne métodos, processos e práticas utilizados na especificação, construção, validação e manutenção de sistemas computacionais (Sommerville, 2011).

Segundo Sommerville (2011), o desenvolvimento de *software* envolve atividades como levantamento de requisitos, modelagem, implementação e testes. A utilização de processos estruturados contribui para que as necessidades identificadas sejam transformadas em funcionalidades e para que o produto desenvolvido possa ser posteriormente verificado em relação aos requisitos estabelecidos.

O levantamento de requisitos constitui uma etapa importante desse processo, pois permite identificar quais funcionalidades o sistema deverá oferecer e quais restrições deverão ser consideradas (Sommerville, 2011). Os requisitos estabelecem uma referência para as etapas posteriores de modelagem, implementação e validação.

A modelagem, por sua vez, possibilita representar diferentes aspectos do sistema antes e durante seu desenvolvimento. Modelos relacionados aos dados, aos atores, às funcionalidades e às relações existentes entre os componentes permitem visualizar a estrutura da aplicação e apoiar a tomada de decisões durante a implementação (Sommerville, 2011).

No desenvolvimento de sistemas *web*, a arquitetura define a forma como os componentes são organizados e como ocorre a comunicação entre eles. A arquitetura cliente-servidor estabelece uma separação entre o componente utilizado pelo usuário para interagir com o sistema e os serviços responsáveis pelo processamento das solicitações e pela aplicação das regras de negócio (Sommerville, 2011). Essa separação permite distribuir responsabilidades e favorecer a manutenção dos componentes.

A persistência dos dados também constitui um elemento importante da arquitetura. As informações utilizadas pela aplicação precisam ser armazenadas de forma que possam manter sua integridade e ser recuperadas posteriormente. Quando o sistema trabalha tanto com dados estruturados quanto com arquivos digitais, diferentes estratégias de armazenamento podem ser adotadas de acordo com as características de cada tipo de informação.

Além da implementação das funcionalidades, a Engenharia de *Software* envolve processos destinados à verificação do comportamento da aplicação. Os testes de *software* permitem executar determinadas funcionalidades e comparar os resultados obtidos com os comportamentos esperados, contribuindo para identificar falhas e verificar regras de negócio (Sommerville, 2011).

Outro aspecto relevante em sistemas *web* é a segurança. Aplicações que armazenam dados e conteúdos pertencentes a diferentes usuários precisam estabelecer mecanismos capazes de identificar quem utiliza o sistema e quais operações cada usuário pode realizar. Autenticação, autorização, validação de entradas e controle de acesso são exemplos de mecanismos que podem ser utilizados para proteger os recursos disponibilizados pela aplicação.

Dessa forma, os princípios da Engenharia de *Software* fornecem a base necessária para transformar as necessidades identificadas em uma solução

computacional estruturada (Sommerville, 2011). O levantamento de requisitos, a modelagem, a definição da arquitetura, a implementação e os testes constituem etapas relacionadas à construção e à validação técnica do *software*.

### 3.8 USABILIDADE E AVALIAÇÃO HEURÍSTICA

Além do funcionamento técnico de uma aplicação, a forma como os usuários interagem com suas funcionalidades constitui um aspecto importante da qualidade do software. Uma aplicação pode executar corretamente suas regras de negócio e, ainda assim, apresentar dificuldades de compreensão, navegação ou utilização (Nielsen, 1994).

A usabilidade está relacionada à facilidade com que os usuários conseguem compreender e utilizar uma interface para realizar as tarefas pretendidas. No contexto de sistemas web, aspectos como clareza das informações, consistência dos elementos, apresentação de mensagens de retorno e prevenção de erros podem influenciar diretamente a interação do usuário com a aplicação (Nielsen, 1994).

Entre os métodos utilizados para avaliar interfaces encontra-se a avaliação heurística, desenvolvida e difundida por Jakob Nielsen. O método consiste na inspeção sistemática de uma interface com base em princípios gerais de usabilidade, denominados heurísticas (Nielsen, 1994). Diferentemente de testes realizados diretamente com usuários, a avaliação heurística permite examinar a interface buscando identificar situações que contrariem princípios reconhecidos de interação. Nielsen é também autor da obra *Usability Engineering*, publicada em 1994, utilizada como referência metodológica neste trabalho.

Nielsen (1994) estabeleceu dez heurísticas gerais para o projeto e a avaliação de interfaces. Esses princípios são denominados heurísticas porque representam orientações gerais de usabilidade, e não regras específicas que determinem uma única forma de construir uma interface.

A primeira heurística, visibilidade do estado do sistema, estabelece que o usuário deve receber informações adequadas sobre o que está acontecendo durante sua interação com a aplicação. A interface deve fornecer retornos em tempo

apropriado, permitindo que o usuário compreenda o resultado das ações realizadas (Nielsen, 1994).

A segunda heurística, correspondência entre o sistema e o mundo real, determina que a interface deve utilizar conceitos, palavras e formas de organização familiares ao usuário, evitando terminologias internas ou excessivamente técnicas quando não forem necessárias (Nielsen, 1994).

A terceira, controle e liberdade do usuário, relaciona-se à possibilidade de interromper ou desfazer ações indesejadas. O usuário pode realizar determinada operação por engano e, nesses casos, a interface deve oferecer meios claros para retornar a um estado anterior ou abandonar a ação (Nielsen, 1994).

A quarta heurística, consistência e padrões, estabelece que elementos semelhantes devem possuir comportamentos e significados semelhantes. A utilização de convenções já conhecidas reduz a necessidade de o usuário aprender novas formas de interação para executar ações equivalentes (Nielsen, 1994).

A quinta, prevenção de erros, considera que o desenvolvimento da interface deve buscar evitar que situações de erro ocorram. Além de apresentar mensagens quando uma falha acontece, o sistema deve utilizar validações, confirmações ou restrições adequadas para reduzir a possibilidade de ações incorretas (Nielsen, 1994).

A sexta heurística, reconhecimento em vez de memorização, busca reduzir a quantidade de informações que o usuário precisa manter na memória durante a navegação. Elementos, opções e ações importantes devem permanecer visíveis ou facilmente acessíveis sempre que forem necessários (Nielsen, 1994).

A sétima, flexibilidade e eficiência de uso, considera que a interface deve atender tanto usuários menos experientes quanto aqueles que possuem maior familiaridade com o sistema. Recursos que tornam determinadas ações mais eficientes podem contribuir para melhorar a utilização frequente da aplicação (Nielsen, 1994).

A oitava heurística, estética e design minimalista, estabelece que as interfaces devem evitar informações irrelevantes ou desnecessárias. A presença excessiva de

elementos pode competir com informações importantes e dificultar a compreensão das ações disponíveis (Nielsen, 1994).

A nona, ajudar os usuários a reconhecer, diagnosticar e recuperar-se de erros, está relacionada à qualidade das mensagens apresentadas quando um problema ocorre. Essas mensagens devem indicar de forma compreensível o que aconteceu e, sempre que possível, orientar o usuário sobre como solucionar a situação (Nielsen, 1994).

Por fim, a décima heurística, ajuda e documentação, considera que, mesmo quando uma aplicação é projetada para ser utilizada de forma intuitiva, podem existir situações nas quais o usuário necessite de orientações adicionais. Nesses casos, as informações de ajuda devem ser acessíveis e direcionadas às tarefas que precisam ser realizadas (Nielsen, 1994). As dez categorias correspondem aos princípios apresentados por Nielsen para avaliação de interfaces.

Neste trabalho, as heurísticas de Nielsen foram utilizadas como critérios de verificação da aderência das interfaces do RepolF aos princípios de usabilidade considerados. Dessa forma, a análise buscou relacionar características observáveis da aplicação a cada uma das dez heurísticas.

No contexto deste trabalho, a utilização das heurísticas de Nielsen complementa a validação técnica realizada por meio dos testes de *software*. Enquanto os testes automatizados verificam aspectos relacionados ao comportamento das funcionalidades e às regras implementadas, a inspeção heurística permite analisar características da interação entre o usuário e a interface.

Assim, a utilização conjunta de testes de *software* e da avaliação heurística permite analisar o RepolF sob perspectivas complementares. Enquanto os testes verificam o comportamento funcional e o atendimento às regras de negócio, a avaliação heurística concentra-se na qualidade da interação entre usuário e interface, fornecendo evidências adicionais para a validação da plataforma desenvolvida.

## 4 METODOLOGIA

Este capítulo apresenta os procedimentos metodológicos empregados na realização da pesquisa, no desenvolvimento do RepolF e na avaliação da solução implementada. O trabalho compreendeu levantamento bibliográfico, definição e refinamento dos requisitos, modelagem, desenvolvimento incremental da aplicação e avaliação do sistema sob duas perspectivas complementares: a verificação do funcionamento técnico, realizada por meio de testes no *backend* e no *frontend*, incluindo testes de integração com banco de dados, e a avaliação da usabilidade das interfaces, que foi conduzida por meio de inspeção heurística baseada nos princípios propostos por Nielsen (1994).

### 4.1 CLASSIFICAÇÃO DA PESQUISA

Quanto à natureza, esta pesquisa caracteriza-se como aplicada, uma vez que teve como finalidade o desenvolvimento de uma solução computacional direcionada ao gerenciamento e ao compartilhamento de materiais didáticos no contexto acadêmico. Em relação aos objetivos, apresenta características exploratórias e descritivas, pois envolveu tanto o estudo dos conceitos relacionados ao problema quanto a descrição da arquitetura, das funcionalidades e dos procedimentos empregados na construção e validação da aplicação.

A abordagem adotada foi predominantemente qualitativa, especialmente na análise dos requisitos, das funcionalidades desenvolvidas e da avaliação heurística de usabilidade. Dados quantitativos provenientes da execução dos testes automatizados também foram utilizados como evidências complementares da validação técnica.

Quanto aos procedimentos, o trabalho combinou pesquisa bibliográfica e desenvolvimento tecnológico. A pesquisa bibliográfica forneceu suporte conceitual para a compreensão de temas relacionados às tecnologias digitais na educação, materiais didáticos digitais, gerenciamento e compartilhamento de informações, sistemas de informação, engenharia de *software* e usabilidade. O desenvolvimento tecnológico compreendeu a especificação, modelagem, implementação e validação do RepolF.

## 4.2 CONTEXTO DA PESQUISA

O Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos, foi adotado como contexto de referência para a definição do problema e do escopo da solução. A escolha está relacionada ao desenvolvimento de uma plataforma direcionada à organização e ao compartilhamento de materiais didáticos em ambiente acadêmico, considerando professores e estudantes como os principais usuários desses recursos.

A pesquisa não utilizou levantamento quantitativo com a comunidade acadêmica, entrevistas ou questionários para determinar a frequência do problema investigado. Sua fundamentação decorreu da literatura relacionada à fragmentação do compartilhamento de informações, à gestão documental e à recuperação de conteúdos em ambientes digitais.

Consequentemente, o trabalho não pretende generalizar seus resultados para todas as instituições de ensino nem afirmar quantitativamente que a utilização do RepolF reduz a perda de arquivos, o retrabalho docente ou melhora o desempenho acadêmico. A avaliação concentra-se na verificação do produto desenvolvido em relação aos requisitos estabelecidos, ao funcionamento técnico das funcionalidades e aos critérios de usabilidade adotados.

## 4.3 PROCESSO DE DESENVOLVIMENTO DO SOFTWARE

O desenvolvimento do RepolF foi conduzido por meio de uma abordagem incremental, na qual a aplicação evoluiu progressivamente a partir da implementação e do refinamento de diferentes conjuntos de funcionalidades.

O processo foi organizado em quatro etapas principais: levantamento e refinamento de requisitos, modelagem, implementação incremental e avaliação.

Na etapa de levantamento e refinamento dos requisitos, foram identificados os comportamentos esperados da aplicação, os diferentes perfis de usuários e as regras de negócio necessárias ao funcionamento da plataforma. Os requisitos foram refinados ao longo do desenvolvimento conforme as necessidades identificadas

durante a construção dos diferentes módulos.

Na etapa de modelagem, foram definidas as entidades e relações responsáveis pela representação das informações do sistema, bem como os principais fluxos de interação entre os usuários e a aplicação. Essa etapa forneceu suporte à estruturação dos dados, às relações entre usuários, disciplinas, materiais, turmas, atividades e demais elementos da plataforma.

A etapa de implementação ocorreu de forma incremental, permitindo a integração progressiva dos módulos da aplicação. O desenvolvimento contemplou tanto os componentes responsáveis pela interface com o usuário quanto aqueles destinados às regras de negócio, persistência das informações, armazenamento de arquivos, autenticação, autorização, administração e moderação.

Por fim, a etapa de avaliação foi dividida em duas frentes. A primeira correspondeu à validação técnica por meio de testes automatizados no *backend* e no *frontend*. A segunda corresponde à avaliação da usabilidade das interfaces por meio de inspeção heurística baseada nos princípios de Nielsen (1994). A utilização dessas duas estratégias permitiu analisar aspectos distintos e complementares da solução. Essas etapas não ocorreram de forma estritamente sequencial, uma vez que o desenvolvimento incremental possibilitou o refinamento contínuo dos requisitos e das funcionalidades ao longo da implementação.

#### 4.4 FONTES DE EVIDÊNCIA E DOCUMENTAÇÃO DO SISTEMA

A descrição técnica do RepolF foi fundamentada prioritariamente na implementação efetivamente desenvolvida. Para isso, foram analisados o código-fonte, as entidades, as migrações do banco de dados, os módulos da aplicação, as regras de negócio, as interfaces e os testes automatizados.

Também foram utilizados como materiais auxiliares a especificação de requisitos do sistema e a documentação referente à modelagem do banco de dados. Esses artefatos contribuíram para manter a documentação apresentada neste trabalho coerente com o comportamento da versão implementada.

A validação técnica utilizou como evidências os casos de teste

implementados e os resultados obtidos durante sua execução. Dessa maneira, procurou-se estabelecer relação entre os requisitos definidos, as funcionalidades desenvolvidas e os cenários utilizados para verificar seu comportamento.

#### 4.5 VALIDAÇÃO TÉCNICA POR TESTES AUTOMATIZADOS

A validação técnica do RepolF foi realizada por meio de testes automatizados no *backend* e no *frontend*, complementados por testes de integração que utilizam uma instância PostgreSQL destinada especificamente ao ambiente de testes.

Os testes foram elaborados para verificar tanto fluxos de execução esperados quanto situações inválidas, tentativas de acesso não autorizado, regras de negócio e condições excepcionais. A utilização de diferentes níveis de teste permitiu avaliar componentes isoladamente e, em determinados cenários, verificar a interação entre a *Application Programming Interface (API)*, suas regras de negócio e a camada de persistência.

##### 4.5.1 Testes automatizados do backend

Os testes automatizados do *backend* foram desenvolvidos utilizando o *Jest*, *framework* de testes que disponibiliza recursos para definição de casos, asserções, *mocks* e execução automatizada das suítes (Jest, [s. d.]). Foram criadas suítes voltadas a diferentes camadas e módulos da aplicação, contemplando *services*, *controllers*, *guards*, validadores e regras de negócio.

Entre os aspectos verificados encontram-se:

- cadastro e autenticação;
- verificação e recuperação de conta;
- validação e expiração de *tokens*;
- controle de acesso conforme papel do usuário;
- autorização baseada na propriedade dos recursos;
- gerenciamento de usuários;
- disciplinas e suas regras de visibilidade;
- turmas, membros e convites;

- atividades e entregas;
- autorização e controle de reenvio;
- gerenciamento e validação de arquivos;
- denúncias e moderação;
- operações administrativas;
- paginação;
- validações de entrada;
- limitação e políticas relacionadas aos uploads;
- segurança dos nomes de arquivos;
- fluxo de confirmação de uploads;
- limpeza de objetos no armazenamento.

Os serviços externos foram substituídos por *mocks* nos testes em que a comunicação efetiva com esses serviços não fazia parte do comportamento que se pretendia verificar. Essa estratégia foi aplicada especialmente às operações relacionadas ao serviço de armazenamento e às notificações por e-mail, evitando a realização de uploads, exclusões ou envios reais durante testes unitários.

O fluxo de *upload* recebeu testes específicos em razão das verificações de segurança implementadas. Foram analisados cenários relacionados à geração e validação da prova assinada de *upload*, sua expiração, alterações indevidas nas informações assinadas e a correspondência entre os dados declarados e o objeto efetivamente armazenado. Também foram testados os mecanismos responsáveis por verificar tamanho e tipo do arquivo antes do registro definitivo dos metadados.

A estratégia persistente de limpeza do armazenamento também foi submetida a testes, verificando o registro de objetos pendentes, a execução das tentativas de exclusão, o tratamento de falhas e a manutenção dos registros que necessitassem de nova tentativa.

#### 4.5.2 Testes automatizados do frontend

Os testes do *frontend* foram desenvolvidos utilizando *Vitest*, em ambiente *jsdom*, juntamente com recursos da *React Testing Library*. O *Vitest* fornece uma estrutura de testes integrada ao ecossistema do *Vite* (*Vitest*, [s. d.]), enquanto a

*React Testing Library* disponibiliza recursos para testar componentes React priorizando comportamentos observáveis em vez de detalhes internos de implementação (Testing Library, [s. d.]).

A abordagem priorizou o comportamento observável pelo usuário em vez da implementação interna dos componentes. Foram simuladas interações como preenchimento de formulários, acionamento de botões, exibição de mensagens, navegação e diferenciação dos recursos apresentados conforme o papel do usuário.

Os cenários contemplaram, entre outros:

- cadastro;
- autenticação;
- verificação de e-mail;
- recuperação e redefinição de senha;
- proteção de rotas;
- permissões específicas de administrador e professor;
- *dashboard*;
- perfil;
- busca de professores;
- criação e edição de disciplinas;
- materiais e políticas de *upload*;
- geração de *Quick Response (QR) Code*;
- turmas;
- ingresso por convite;
- atividades;
- envio e reenvio de entregas;
- páginas administrativas;
- gerenciamento de usuários;
- gerenciamento de arquivos;
- moderação de denúncias;
- mensagens de confirmação;
- notificações;
- comportamento do cliente *Hypertext Transfer Protocol (HTTP)*;
- elementos relacionados à acessibilidade;

Também foram verificadas situações de erro e recuperação, incluindo convites inválidos, falhas na obtenção de informações e disponibilização de opções para repetição de determinadas operações.

#### 4.5.3 Testes de integração com PostgreSQL

Além dos testes executados com dependências isoladas ou simuladas, foram desenvolvidos testes de integração utilizando uma instância real do PostgreSQL destinada exclusivamente ao ambiente de testes.

Essa etapa teve como objetivo verificar comportamentos que dependem diretamente das características do banco relacional e que poderiam não ser reproduzidos adequadamente com *repositories* simulados.

Foram criados cenários relacionados a:

- execução das migrações;
- autenticação por meio de requisições *HTTP*;
- administração;
- disciplinas;
- turmas;
- atividades;
- operações que dependem de restrições de integridade;

Um banco PostgreSQL isolado foi utilizado para essas execuções, evitando a utilização dos dados do ambiente de desenvolvimento ou produção. As suítes dependentes desse ambiente foram configuradas para permanecer desabilitadas quando a variável de conexão com o banco de testes não estivesse disponível.

Essa separação permitiu que a suíte padrão permanecesse executável independentemente da existência de uma instância local de PostgreSQL, enquanto os testes de integração puderam ser executados especificamente quando o ambiente necessário estivesse configurado.

Os testes envolvendo banco real também permitiram verificar comportamentos relacionados às restrições de unicidade e a condições de concorrência, como tentativas de criação de contas duplicadas, duplicidade de

vínculos e submissões e registro simultâneo de denúncias pendentes para o mesmo alvo.

#### 4.6 AVALIAÇÃO DE USABILIDADE POR INSPEÇÃO HEURÍSTICA

Além da validação técnica realizada por meio dos testes automatizados e de integração, a usabilidade do RepolF foi analisada por meio de uma inspeção heurística baseada nas dez heurísticas de usabilidade propostas por Nielsen (1994). A avaliação teve como finalidade verificar a aderência das principais interfaces e fluxos de interação da plataforma aos princípios estabelecidos pelo autor.

A inspeção foi conduzida pelo próprio autor e abrangeu interfaces representativas das funcionalidades disponíveis aos diferentes perfis do sistema, incluindo áreas de acesso público e recursos destinados a estudantes, professores e administradores. A análise considerou situações relacionadas à autenticação e ao cadastro, à consulta e ao gerenciamento de disciplinas e materiais, à participação e ao gerenciamento de turmas, às atividades e entregas e às funcionalidades administrativas.

A avaliação foi orientada pelas dez heurísticas de Nielsen: visibilidade do estado do sistema; correspondência entre o sistema e o mundo real; controle e liberdade do usuário; consistência e padrões; prevenção de erros; reconhecimento em vez de memorização; flexibilidade e eficiência de uso; estética e *design* minimalista; auxílio no reconhecimento, diagnóstico e recuperação de erros; e ajuda e documentação.

Para cada heurística, foram examinados elementos e comportamentos das interfaces capazes de evidenciar sua aplicação no RepolF. A análise considerou aspectos como os mecanismos de *feedback* apresentados após as operações, a terminologia utilizada, a organização e padronização dos componentes, as possibilidades de cancelamento e confirmação de ações, as validações empregadas para prevenir erros, a apresentação contextual das informações, os recursos destinados a facilitar tarefas recorrentes e a forma como mensagens de erro e orientações são disponibilizadas aos usuários.

A inspeção foi realizada de maneira sistemática, confrontando as

características observadas nas interfaces selecionadas com os princípios estabelecidos por Nielsen (1994). A análise de cada heurística foi fundamentada em evidências concretas obtidas na própria aplicação, evitando que a avaliação se limitasse a uma apreciação exclusivamente subjetiva da interface.

Os resultados foram apresentados individualmente para cada uma das dez heurísticas. Para cada princípio, foi descrita sua aplicação no RepolF e apresentada uma evidência visual representativa da interface ou do comportamento analisado. Dessa forma, a avaliação foi estruturada no formato de heurística, análise textual e evidência visual, permitindo relacionar diretamente cada princípio de usabilidade às características observadas na plataforma.

As evidências visuais foram obtidas por meio de capturas das próprias interfaces do RepolF, selecionadas de acordo com sua capacidade de demonstrar os princípios analisados. Uma mesma interface pôde fornecer evidências para diferentes heurísticas, desde que os aspectos considerados fossem discutidos individualmente em cada análise.

#### 4.7 PARTICIPANTES E ASPECTOS ÉTICOS

A pesquisa não envolveu aplicação de questionários, entrevistas, grupos focais ou experimentos estruturados com estudantes, professores ou outros participantes humanos. A solução foi avaliada por meio de testes técnicos e de inspeção heurística conduzida pelo próprio autor.

Por esse motivo, não foram coletados dados de participantes para análise científica, tampouco foram utilizadas respostas individuais ou informações comportamentais de usuários para a obtenção dos resultados apresentados.

Os dados utilizados durante o desenvolvimento e a execução dos testes foram tratados como dados de teste, evitando a exposição de informações pessoais reais na documentação, nas capturas de tela e nas demais evidências apresentadas.

A ausência de participantes humanos também delimita as conclusões que podem ser obtidas. Os resultados permitem analisar o atendimento aos requisitos, o

funcionamento técnico da solução e a aderência das interfaces aos princípios de usabilidade considerados na inspeção heurística, mas não permitem determinar o grau de satisfação da comunidade acadêmica, a aceitação da plataforma por seus usuários ou seu impacto sobre o processo de ensino e aprendizagem.

## 5 REQUISITOS E MODELAGEM DO SISTEMA

Este capítulo apresenta os requisitos e os modelos que representam o funcionamento do RepolF. São descritos o escopo da solução, os atores, os requisitos funcionais e não funcionais, as regras de negócio, os casos de uso e a modelagem dos dados persistidos pela aplicação. Os modelos apresentados foram elaborados a partir das funcionalidades implementadas e da estrutura do sistema.

### 5.1 ESCOPO DO SISTEMA

O RepolF foi desenvolvido para centralizar a organização e o compartilhamento de materiais didáticos em ambiente acadêmico. A plataforma permite que professores criem disciplinas, disponibilizem materiais, organizem turmas e publiquem atividades, enquanto estudantes podem acessar conteúdos relacionados às turmas das quais participam e realizar entregas de trabalhos.

A solução também contempla recursos de autenticação, recuperação de conta, controle de acesso por perfil, administração de usuários e arquivos, registro de denúncias e moderação da plataforma.

O sistema possui três perfis principais de usuários: professor, estudante e administrador.

O professor é responsável pela criação e pelo gerenciamento de disciplinas, materiais, turmas e atividades. O estudante acessa os conteúdos disponibilizados, participa de turmas e realiza entregas de atividades. O administrador gerencia usuários, arquivos, denúncias e informações gerais da plataforma.

Apesar de possuir funcionalidades relacionadas a turmas e atividades acadêmicas, o RepolF não pretende substituir integralmente um sistema acadêmico ou um Ambiente Virtual de Aprendizagem. A plataforma não realiza controle de notas, frequência, matrícula institucional ou comunicação síncrona entre usuários. Seu escopo permanece concentrado na organização, no compartilhamento, na recuperação e no controle de acesso aos materiais didáticos, incorporando funcionalidades complementares necessárias para contextualizar esses conteúdos.

## 5.2 ATORES DO SISTEMA

Os atores representam os usuários e serviços externos que interagem com o RepolF. O Quadro 1 apresenta os atores identificados e suas responsabilidades gerais.

Quadro 1 - Atores do RepolF

| Ator                     | Descrição                                                                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Visitante                | Usuário que ainda não está autenticado. Pode realizar cadastro, autenticação, verificação de e-mail e recuperação de senha.                         |
| Estudante                | Usuário autenticado que pode acessar seu perfil, participar de turmas, consultar disciplinas vinculadas, visualizar materiais e enviar atividades.  |
| Professor                | Usuário autenticado responsável por criar e gerenciar disciplinas, materiais, turmas, atividades, convites e entregas dos estudantes.               |
| Administrador            | Usuário autenticado com acesso às funções administrativas, incluindo gerenciamento de usuários, arquivos, denúncias e estatísticas.                 |
| Serviço de armazenamento | Serviço externo responsável pelo armazenamento dos arquivos digitais, pela geração de endereços temporários e pela remoção dos objetos armazenados. |
| Serviço de e-mail        | Serviço externo responsável pelo envio de mensagens de verificação de conta, recuperação de senha e notificações acadêmicas.                        |

Fonte: autor (2026).

## 5.3 REQUISITOS FUNCIONAIS

Os requisitos funcionais descrevem os serviços e comportamentos que o RepolF deve disponibilizar. Na sequência, são apresentados os requisitos funcionais especificados para a plataforma, organizados em quadros e identificados pelo prefixo RF, seguido de uma numeração sequencial. O Quadro 2 apresenta os requisitos relacionados à autenticação e ao perfil dos usuários; o Quadro 3 reúne os requisitos de disciplinas e materiais; o Quadro 4 apresenta os requisitos referentes

às turmas; o Quadro 5 contempla as funcionalidades de atividades e entregas; e o Quadro 6 reúne os requisitos de administração e moderação.

Quadro 2 - Requisitos funcionais de autenticação e perfil

| Identificador | Requisito funcional                                                                                                                                                                                    |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RF01          | O sistema deve permitir o cadastro de contas com os perfis de estudante ou professor, sendo a criação de contas administrativas restrita às funcionalidades de administração.                          |
| RF02          | O sistema deve enviar uma mensagem de verificação para o e-mail informado durante o cadastro e impedir a autenticação enquanto a conta não estiver verificada.                                         |
| RF03          | O sistema deve permitir a autenticação por nome de usuário ou endereço de e-mail, mediante fornecimento da senha correspondente.                                                                       |
| RF04          | O sistema deve permitir a solicitação e a conclusão da recuperação de senha por meio de token temporário enviado por e-mail.                                                                           |
| RF05          | O sistema deve permitir que o usuário autenticado altere seu nome de exibição e sua senha.                                                                                                             |
| RF06          | O sistema deve disponibilizar o perfil do usuário, apresentando informações e indicadores de atividade de acordo com seu papel.                                                                        |
| RF07          | O sistema deve permitir a listagem, a busca e a consulta de perfis de professores, respeitando as regras de acesso relacionadas ao perfil do usuário solicitante e aos vínculos acadêmicos existentes. |
| RF08          | O sistema deve encerrar a sessão quando o <i>token</i> de autenticação atingir seu prazo de expiração.                                                                                                 |

Fonte: autor (2026).

Quadro 3 - Requisitos funcionais de disciplinas e materiais

| Identificador | Requisito funcional                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------|
| RF09          | O sistema deve permitir que professores criem disciplinas informando nome, descrição e visibilidade (pública/privada). |
| RF10          | O sistema deve permitir que o professor liste e pesquise suas disciplinas.                                             |
| RF11          | O sistema deve permitir que o professor edite ou exclua apenas as disciplinas sob sua responsabilidade.                |
| RF12          | O sistema deve permitir que uma disciplina seja definida como                                                          |

|      |                                                                                                                                                                                                                                                                                                                                                                                                |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      | pública ou privada.                                                                                                                                                                                                                                                                                                                                                                            |
| RF13 | O sistema deve permitir que professores solicitem uma <i>Uniform Resource Locator (URL)</i> temporária para envio de um arquivo vinculado a uma disciplina de sua propriedade.                                                                                                                                                                                                                 |
| RF14 | O sistema deve registrar no banco de dados os metadados do arquivo enviado ao serviço de armazenamento.                                                                                                                                                                                                                                                                                        |
| RF15 | O sistema deve permitir a listagem e a pesquisa de materiais pelo nome do arquivo, respeitando sua visibilidade e seu estado.                                                                                                                                                                                                                                                                  |
| RF16 | O sistema deve permitir o acesso e o download de materiais públicos por qualquer pessoa que possua o respectivo link, mesmo sem autenticação. Materiais privados devem permanecer restritos ao professor proprietário nas operações convencionais da plataforma, podendo ser compartilhados intencionalmente pelo proprietário por meio de URL temporária pré-assinada, inclusive via QR Code. |
| RF17 | O sistema deve permitir que o professor altere o nome e a visibilidade dos arquivos sob sua propriedade.                                                                                                                                                                                                                                                                                       |
| RF18 | O sistema deve permitir que o professor desabilite e reabilite um arquivo de forma reversível.                                                                                                                                                                                                                                                                                                 |
| RF19 | O sistema deve permitir a exclusão definitiva de um arquivo, removendo seu registro e promovendo a remoção do objeto correspondente no serviço de armazenamento, com processamento posterior quando essa remoção não puder ser concluída imediatamente.                                                                                                                                        |

Fonte: autor (2026).

Quadro 4 - Requisitos funcionais de turmas

| Identificador | Requisito funcional                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------|
| RF20          | O sistema deve permitir que professores criem, listem, pesquisem, editem e excluam suas turmas.                 |
| RF21          | O sistema deve permitir que o professor vincule à turma uma disciplina já existente sob sua responsabilidade.   |
| RF22          | O sistema deve permitir que professores adicionem estudantes diretamente à turma por nome de usuário ou e-mail. |
| RF23          | O sistema deve permitir que professores consultem e removam os membros ativos da turma.                         |
| RF24          | O sistema deve permitir que professores gerem convites temporários para ingresso na turma.                      |

|      |                                                                                                         |
|------|---------------------------------------------------------------------------------------------------------|
| RF25 | O sistema deve permitir que estudantes solicitem ingresso em uma turma por meio de um convite válido.   |
| RF26 | O sistema deve permitir que o professor consulte, aceite ou rejeite solicitações pendentes de ingresso. |
| RF27 | O sistema deve permitir que professores e estudantes consultem as turmas às quais possuem acesso.       |

Fonte: autor (2026).

Quadro 5 - Requisitos funcionais de atividades e entregas

| Identificador | Requisito funcional                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| RF28          | O sistema deve permitir que professores criem atividades vinculadas às suas disciplinas.                                                      |
| RF29          | O sistema deve permitir que a atividade contenha título, descrição, prazo e anexo opcional.                                                   |
| RF30          | O sistema deve permitir que o professor edite ou exclua atividades sob sua responsabilidade.                                                  |
| RF31          | O sistema deve permitir que professores e estudantes autorizados visualizem as atividades de uma disciplina.                                  |
| RF32          | O sistema deve permitir que estudantes pertencentes a uma turma relacionada à disciplina enviem suas entregas antes do encerramento do prazo. |
| RF33          | O sistema deve permitir que o estudante consulte sua própria entrega e seu respectivo estado.                                                 |
| RF34          | O sistema deve permitir que o professor consulte os estudantes que realizaram e os que ainda não realizaram a entrega.                        |
| RF35          | O sistema deve permitir que o professor faça o <i>download</i> das entregas recebidas.                                                        |
| RF36          | O sistema deve permitir que o professor autorize individualmente o reenvio de uma entrega.                                                    |
| RF37          | O sistema deve substituir o arquivo anterior quando o estudante realizar um reenvio autorizado.                                               |

Fonte: autor (2026).

Quadro 6 - Requisitos funcionais de administração e moderação

| Identificador | Requisito funcional |
|---------------|---------------------|
|---------------|---------------------|

|      |                                                                                                                                  |
|------|----------------------------------------------------------------------------------------------------------------------------------|
| RF38 | O sistema deve permitir que usuários autenticados denunciem usuários ou arquivos.                                                |
| RF39 | O sistema deve permitir que o usuário consulte as denúncias registradas por sua conta.                                           |
| RF40 | O sistema deve permitir que administradores consultem estatísticas referentes a usuários, arquivos e denúncias.                  |
| RF41 | O sistema deve permitir que administradores listem, pesquisem e filtrem usuários.                                                |
| RF42 | O sistema deve permitir que administradores criem contas previamente verificadas.                                                |
| RF43 | O sistema deve permitir que administradores alterem o papel de outros usuários.                                                  |
| RF44 | O sistema deve permitir que administradores excluam logicamente e restaurem contas de usuários.                                  |
| RF45 | O sistema deve permitir que administradores listem, pesquisem e visualizem arquivos públicos, privados ou excluídos logicamente. |
| RF46 | O sistema deve permitir que administradores realizem a exclusão lógica, a restauração ou a exclusão definitiva de arquivos.      |
| RF47 | O sistema deve permitir que administradores listem e filtrem denúncias por situação e tipo de alvo.                              |
| RF48 | O sistema deve permitir que administradores alterem a situação de uma denúncia e registrem uma observação de resolução.          |

Fonte: autor (2026).

## 5.4 REQUISITOS NÃO FUNCIONAIS

O Quadro 7 apresenta os principais requisitos não funcionais (RNF) relacionados à arquitetura, segurança, desempenho e usabilidade do RepolF.

Quadro 7 - Requisitos não funcionais

| Identificador | Requisito não funcional                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------|
| RNF01         | A aplicação deve ser acessível por navegadores <i>web</i> e adaptar sua interface a diferentes dimensões de tela. |
| RNF02         | A aplicação deve apresentar navegação, mensagens e elementos de interface em linguagem compreensível ao usuário.  |

|       |                                                                                                                                                           |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| RNF03 | A comunicação entre <i>frontend</i> e <i>backend</i> deve ocorrer por meio de uma <i>API Representational State Transfer</i> (REST).                      |
| RNF04 | O <i>backend</i> deve manter organização modular, separando responsabilidades relacionadas aos diferentes domínios da aplicação.                          |
| RNF05 | O sistema deve utilizar autenticação baseada em <i>tokens JWT</i> para identificar usuários autenticados.                                                 |
| RNF06 | O sistema deve aplicar controle de acesso conforme os papéis de administrador, professor e estudante.                                                     |
| RNF07 | As senhas devem ser armazenadas por meio de função de <i>hash</i> , sem persistência em texto simples.                                                    |
| RNF08 | As entradas recebidas pela <i>API</i> devem ser validadas antes da execução das operações.                                                                |
| RNF09 | O sistema deve aplicar limitação de requisições para reduzir tentativas excessivas de autenticação e envio de mensagens.                                  |
| RNF10 | A <i>API</i> deve utilizar mecanismos de proteção de cabeçalhos <i>HTTP</i> e restrição de origens permitidas.                                            |
| RNF11 | Os arquivos binários devem ser armazenados em serviço de armazenamento de objetos, enquanto seus metadados devem permanecer no banco de dados relacional. |
| RNF12 | O sistema deve impedir o envio de arquivos maiores que 200 MB.                                                                                            |
| RNF13 | O sistema deve bloquear extensões associadas a executáveis e instaladores potencialmente perigosos.                                                       |
| RNF14 | O banco de dados deve utilizar identificadores <i>Universally Unique Identifier</i> (UUID) para as entidades persistidas.                                 |
| RNF15 | Listagens extensas devem possuir paginação e, quando aplicável, mecanismos de busca ou filtragem.                                                         |
| RNF16 | Usuários e arquivos devem possuir exclusão lógica quando houver necessidade de restauração posterior.                                                     |
| RNF17 | Falhas em serviços de notificação não devem desfazer operações acadêmicas que já tenham sido concluídas com sucesso.                                      |
| RNF18 | A aplicação deve apresentar mensagens de carregamento, sucesso e erro durante as principais operações.                                                    |

Fonte: autor (2026).

## 5.5 REGRAS DE NEGÓCIO

As regras de negócio estabelecem restrições e condições que determinam como as funcionalidades devem se comportar. Para sua identificação, foi utilizado o prefixo RN.

### 5.5.1 Contas e autenticação

Quadro 8 - Regras de negócio de contas e autenticação

| Identificador | Regra de negócio                                                                                                       |
|---------------|------------------------------------------------------------------------------------------------------------------------|
| RN01          | O cadastro realizado pelo próprio usuário deve permitir somente os perfis de estudante e professor.                    |
| RN02          | Contas administrativas devem ser criadas ou definidas por um administrador autorizado.                                 |
| RN03          | O nome de usuário e o endereço de e-mail devem ser únicos.                                                             |
| RN04          | A conta somente poderá ser autenticada depois da confirmação do endereço de e-mail.                                    |
| RN05          | O <i>token</i> de verificação de e-mail deve possuir validade de 24 horas.                                             |
| RN06          | O <i>token</i> de recuperação de senha deve possuir validade de 10 minutos e ser utilizado uma única vez.              |
| RN07          | A senha deve possuir, no mínimo, oito caracteres.                                                                      |
| RN08          | A alteração da senha por usuário autenticado deve exigir a senha atual, e a nova senha deve ser diferente da anterior. |
| RN09          | O nome de usuário representa a identidade utilizada nas rotas de perfil e não pode ser alterado pelo próprio usuário.  |

Fonte: autor (2026).

### 5.5.2 Disciplinas e materiais

Quadro 9 - Regras de negócio de disciplinas e materiais

| Identificador | Regra de negócio                                                                 |
|---------------|----------------------------------------------------------------------------------|
| RN10          | Somente professores podem criar disciplinas e publicar materiais.                |
| RN11          | O professor somente pode editar ou excluir disciplinas sob sua responsabilidade. |

|      |                                                                                                                                                                                                                                                                                                                                                                                                            |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RN12 | Uma disciplina privada pode ser acessada pelo professor proprietário e pelos estudantes ativos de turmas às quais ela esteja vinculada.                                                                                                                                                                                                                                                                    |
| RN13 | Um arquivo privado somente pode ser visualizado ou baixado pelo professor proprietário, não sendo disponibilizado automaticamente aos estudantes mesmo quando estiver vinculado a uma disciplina associada a uma turma. O proprietário pode, entretanto, compartilhá-lo intencionalmente por meio de URL temporária pré-assinada, inclusive via QR Code, durante o período de validade do endereço gerado. |
| RN14 | A exclusão definitiva deve remover permanentemente o registro do arquivo e promover a remoção do objeto correspondente no serviço de armazenamento. Quando a remoção do objeto não puder ser concluída imediatamente nos fluxos que utilizam o mecanismo de limpeza, sua chave deve permanecer registrada para nova tentativa.                                                                             |
| RN15 | O <i>upload</i> deve aceitar arquivos com tamanho máximo de 200 MB.                                                                                                                                                                                                                                                                                                                                        |
| RN16 | Nomes terminados em extensões executáveis ou de instalação bloqueadas não devem ser aceitos.                                                                                                                                                                                                                                                                                                               |
| RN17 | A <i>URL</i> pré-assinada para envio deve possuir duração limitada e estar associada ao tipo e ao tamanho declarado do arquivo.                                                                                                                                                                                                                                                                            |

Fonte: autor (2026).

### 5.5.3 Turmas

Quadro 10 - Regras de negócio de turmas

| Identificador | Regra de negócio                                                                  |
|---------------|-----------------------------------------------------------------------------------|
| RN18          | Somente professores podem criar e administrar turmas.                             |
| RN19          | O professor somente pode editar ou excluir turmas sob sua propriedade.            |
| RN20          | O professor somente pode vincular à turma disciplinas pertencentes à sua conta.   |
| RN21          | Um estudante pode possuir somente um vínculo com cada turma.                      |
| RN22          | O vínculo do estudante pode assumir os estados ativo ou pendente.                 |
| RN23          | Estudantes com vínculo pendente não podem acessar os conteúdos internos da turma. |
| RN24          | O ingresso por convite deve ser permitido somente para contas com                 |

|      |                                                                                                      |
|------|------------------------------------------------------------------------------------------------------|
|      | perfil de estudante.                                                                                 |
| RN25 | O ingresso por convite deve criar uma solicitação pendente, que dependerá da aceitação do professor. |
| RN26 | O convite deve possuir prazo de validade escolhido entre as opções disponibilizadas pelo sistema.    |

Fonte: autor (2026).

#### 5.5.4 Atividades e entregas

Quadro 11 - Regras de negócio de atividades e entregas

| Identificador | Regra de negócio                                                                                              |
|---------------|---------------------------------------------------------------------------------------------------------------|
| RN27          | O professor somente pode criar atividades em disciplinas sob sua responsabilidade.                            |
| RN28          | A data limite de uma nova atividade deve ser, no mínimo, o dia seguinte à data de criação.                    |
| RN29          | Somente estudantes ativos em uma turma vinculada à disciplina podem realizar entregas.                        |
| RN30          | A entrega deve ocorrer antes do encerramento do prazo definido para a atividade.                              |
| RN31          | Cada estudante deve possuir, no máximo, uma entrega registrada por atividade.                                 |
| RN32          | Um estudante que já realizou a entrega somente poderá reenviá-la após autorização do professor.               |
| RN33          | Depois de um reenvio autorizado, a permissão de reenvio deve retornar automaticamente ao estado desabilitado. |
| RN34          | O reenvio deve substituir o arquivo anteriormente entregue e atualizar a data da submissão.                   |
| RN35          | O professor responsável pela atividade pode consultar e baixar todas as entregas.                             |

Fonte: autor (2026).

#### 5.5.5 Administração e denúncias

Quadro 12 - Regras de negócio de administração e denúncias

| Identificador | Regra de negócio                                                                                   |
|---------------|----------------------------------------------------------------------------------------------------|
| RN36          | O administrador não pode alterar o próprio papel.                                                  |
| RN37          | O administrador não pode excluir a própria conta.                                                  |
| RN38          | Contas criadas pelo administrador devem ser cadastradas como verificadas.                          |
| RN39          | A exclusão lógica de um usuário ou arquivo deve permitir sua restauração posterior.                |
| RN40          | A exclusão definitiva de um arquivo não deve permitir restauração.                                 |
| RN41          | O administrador pode visualizar arquivos privados ou excluídos logicamente para fins de moderação. |
| RN42          | Uma denúncia deve possuir como alvo um usuário ou um arquivo existente.                            |
| RN43          | O usuário não pode denunciar a própria conta.                                                      |
| RN44          | Um mesmo usuário não pode manter duas denúncias pendentes para o mesmo alvo.                       |
| RN45          | Uma denúncia pode possuir os estados pendente, em análise, resolvida ou descartada.                |
| RN46          | O responsável pela resolução deve ser registrado quando a denúncia for resolvida ou descartada.    |

Fonte: autor (2026).

## 5.6 MODELAGEM DE CASOS DE USO

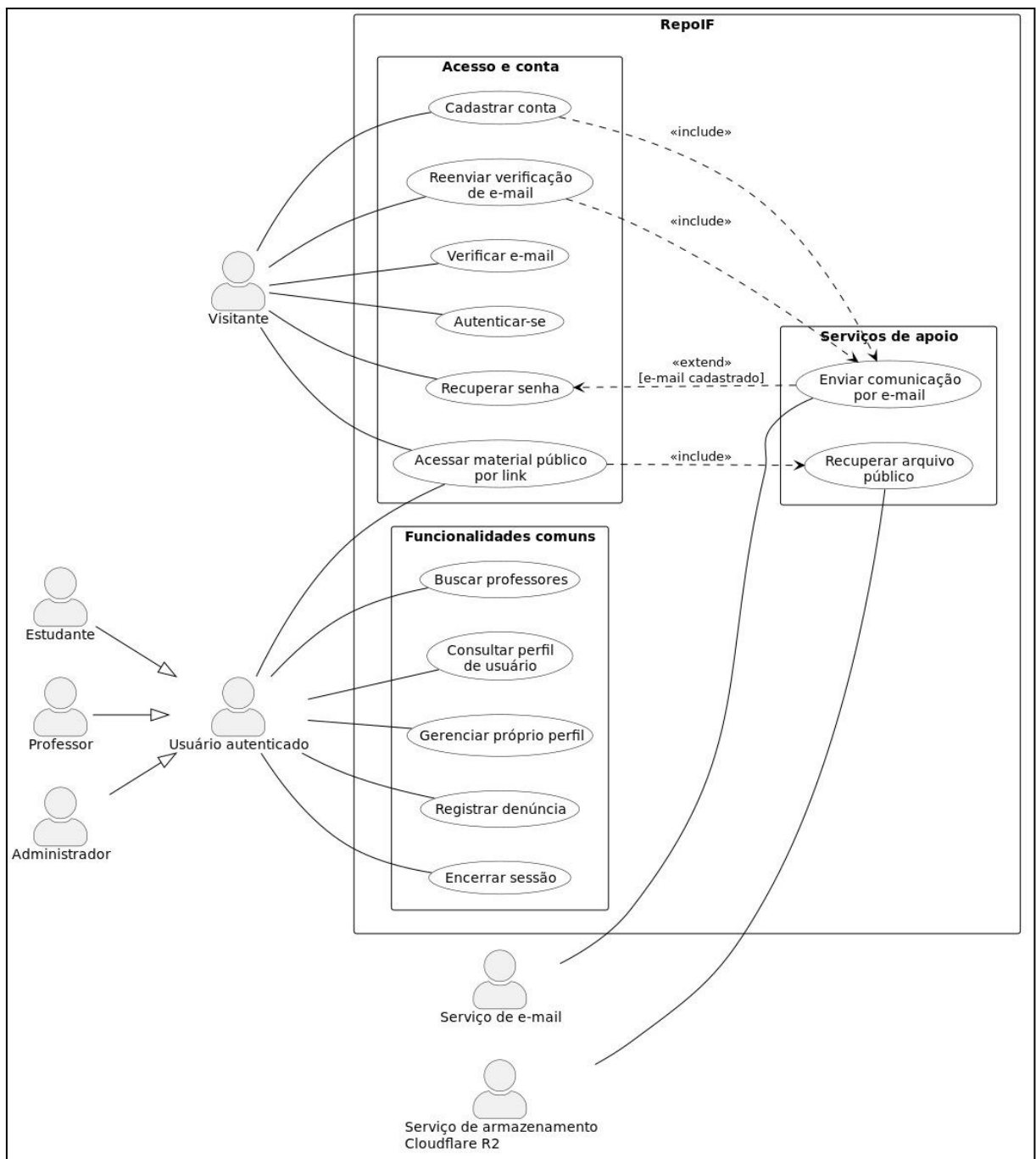
Os casos de uso do RepolF foram organizados conforme os diferentes perfis de interação com a plataforma. Essa divisão foi adotada devido à quantidade de funcionalidades existentes e às diferenças entre as permissões atribuídas a estudantes, professores e administradores, permitindo preservar a legibilidade dos modelos. Inicialmente são apresentadas as funcionalidades relacionadas ao acesso e às operações comuns aos usuários autenticados. Em seguida, são detalhados os casos de uso específicos de estudante, professor e administrador.

### 5.6.1 Acesso, conta e funcionalidades comuns

A Figura 1 apresenta os casos de uso relacionados ao acesso à plataforma e

às funcionalidades comuns aos usuários autenticados, incluindo cadastro, autenticação, gerenciamento da conta e operações compartilhadas entre os diferentes perfis.

Figura 1 - Casos de uso de acesso, conta e funcionalidades comuns do RepolF

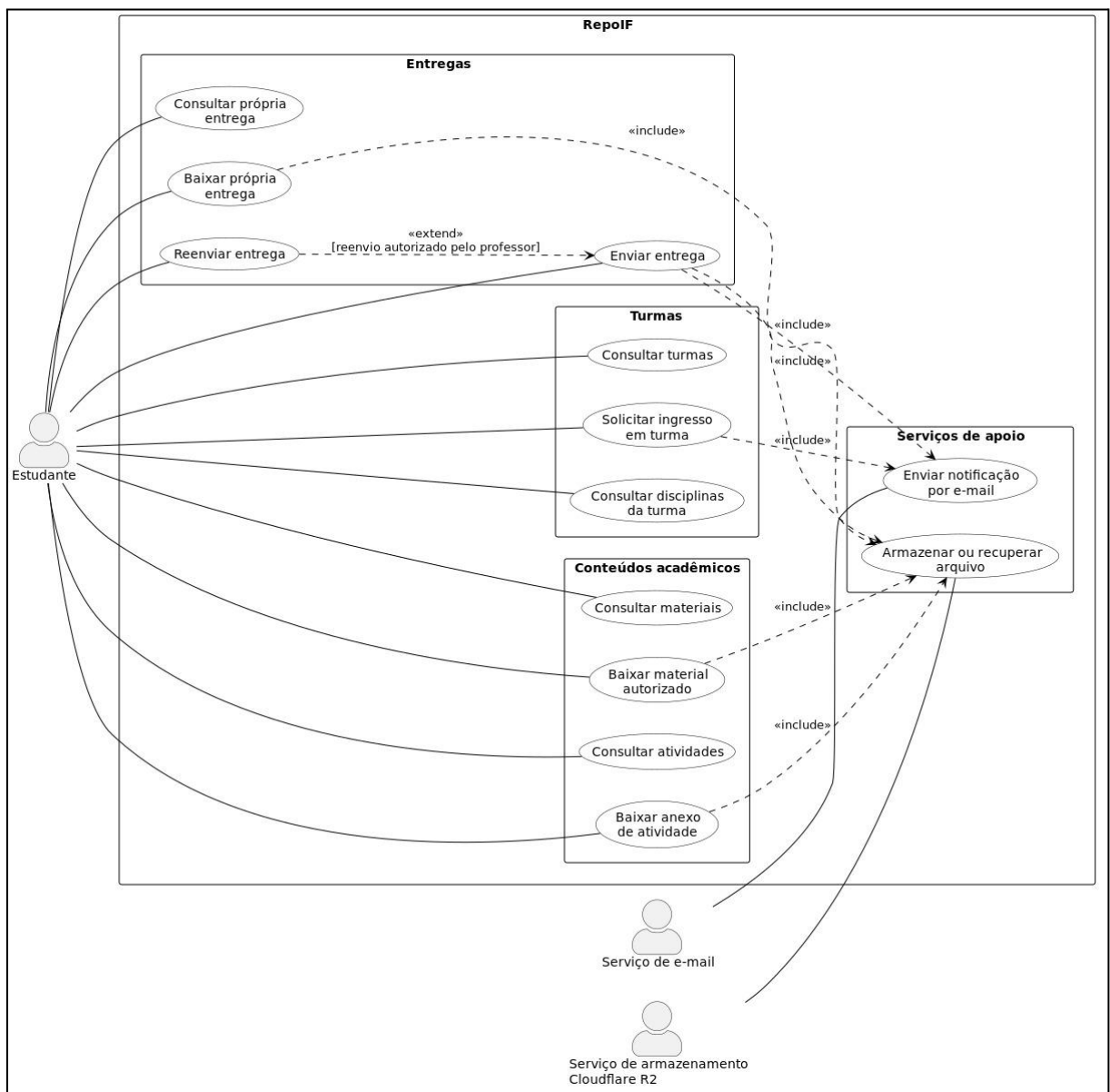


Fonte: autor (2026).

### 5.6.2 Casos de uso do estudante

A Figura 2 representa as funcionalidades disponibilizadas ao estudante, contemplando operações relacionadas à participação em turmas, acesso aos conteúdos acadêmicos e envio de atividades, respeitando as permissões definidas pelo sistema.

Figura 2 - Casos de uso do estudante no RepolF

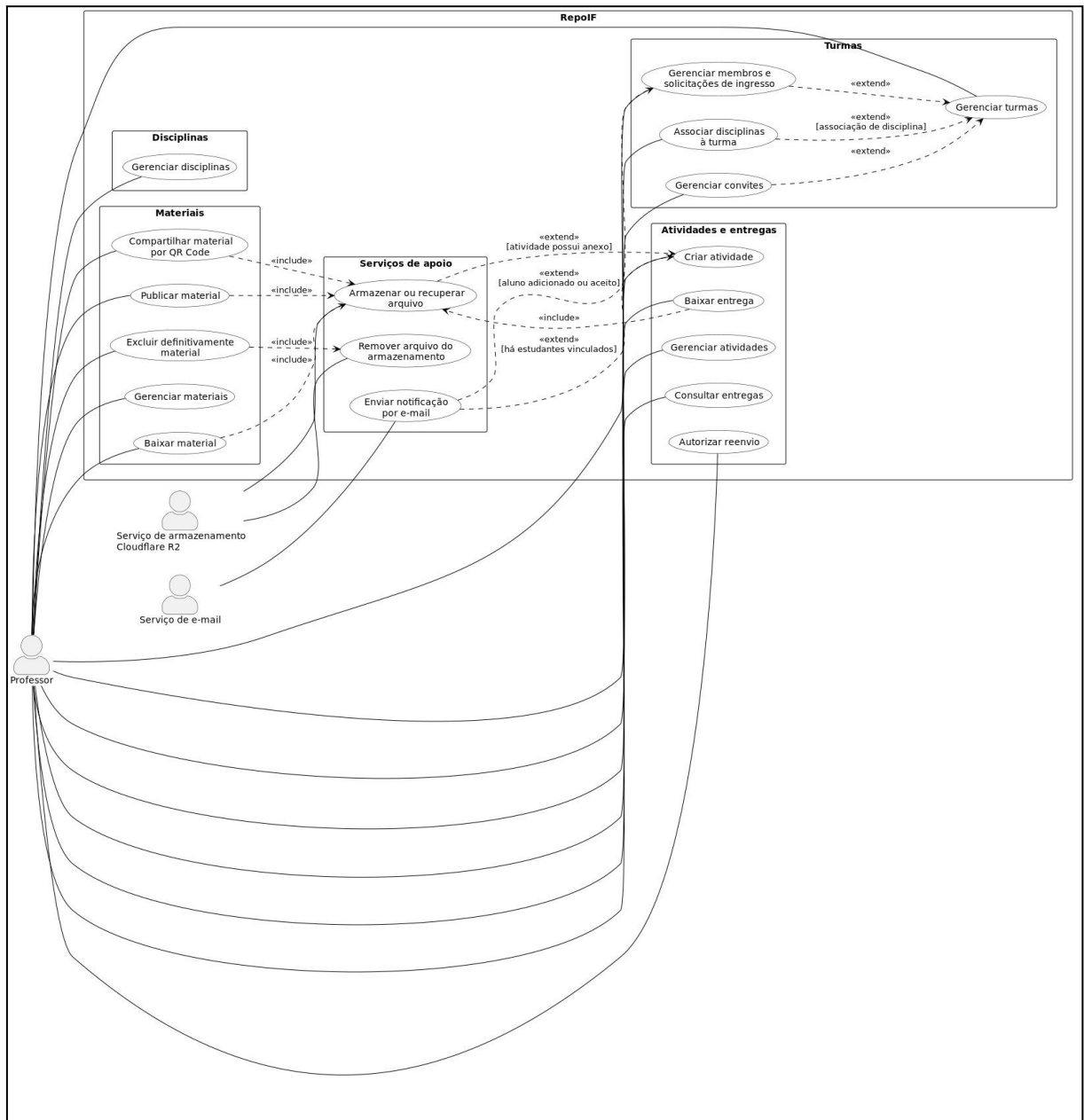


Fonte: autor (2026).

### 5.6.3 Casos de uso do professor

A Figura 3 ilustra as operações executadas pelo professor, responsáveis pelo gerenciamento de disciplinas, materiais, turmas e atividades, além do acompanhamento das entregas realizadas pelos estudantes.

Figura 3 - Casos de uso do professor no RepoIF



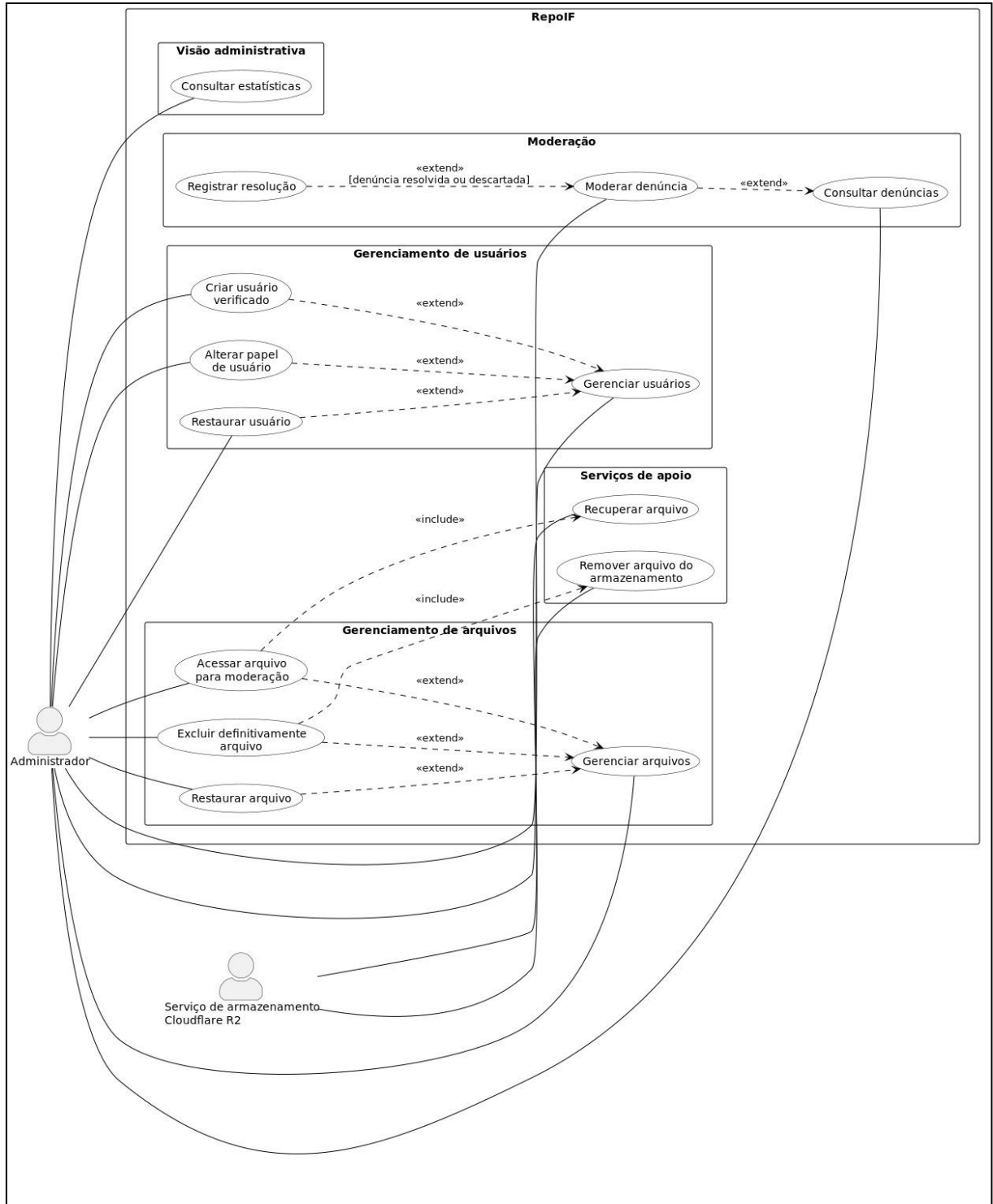
Fonte: autor (2026).

#### 5.6.4 Casos de uso do administrador

A Figura 4 reúne as funcionalidades administrativas da plataforma, responsáveis pelas atividades de administração e moderação da plataforma,

incluindo o gerenciamento de usuários, arquivos, denúncias e informações gerais do sistema.

Figura 4 - Casos de uso do administrador no RepolF

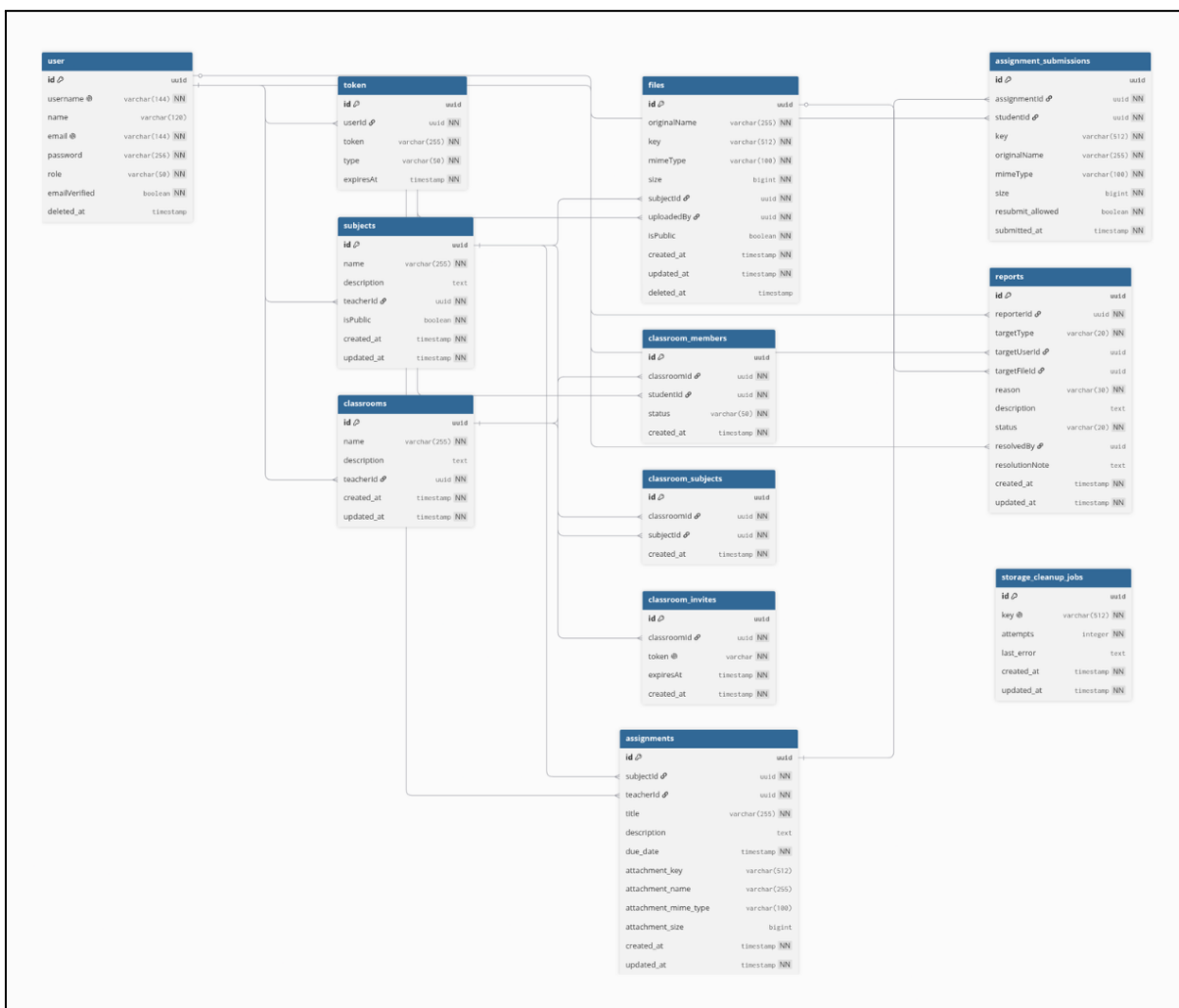


Fonte: autor (2026).

## 5.7 MODELAGEM DOS DADOS

O RepolF utiliza banco de dados relacional para armazenar as informações estruturadas da aplicação, enquanto os arquivos binários permanecem em um serviço de armazenamento de objetos. As entidades utilizam identificadores *UUID*, permitindo a identificação única dos registros.

Figura 5 - Modelo Entidade-Relacionamento do RepolF



Fonte: autor (2026).

Conforme apresentado na Figura 5, a entidade de usuários relaciona-se a diferentes módulos da plataforma, representando os papéis desempenhados por professores, estudantes e administradores. As relações entre turmas e estudantes e entre turmas e disciplinas são representadas por entidades associativas, permitindo relacionamentos muitos para muitos. As atividades são vinculadas às disciplinas e aos professores responsáveis, enquanto as entregas associam cada atividade ao

estudante que realizou a submissão. Os arquivos armazenados no *Cloudflare R2* são representados no banco por seus metadados, preservando a separação entre o armazenamento dos objetos binários e a persistência das informações estruturadas da aplicação.

### 5.7.1 Entidades do sistema

Quadro 13 - Entidades persistidas pelo RepolF

| Entidade                  | Finalidade                                                                                        | Principais atributos                                                                       |
|---------------------------|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| <i>user</i>               | Armazena as contas cadastradas na plataforma.                                                     | <i>id, username, name, email, password, role, emailVerified e deletedAt.</i>               |
| <i>token</i>              | Armazena <i>tokens</i> temporários utilizados na verificação de e-mail e na recuperação de senha. | <i>id, userId, token, type e expiresAt.</i>                                                |
| <i>subjects</i>           | Representa as disciplinas criadas pelos professores.                                              | <i>id, name, description, teacherId, isPublic, createdAt e updatedAt.</i>                  |
| <i>files</i>              | Armazena os metadados dos materiais enviados ao armazenamento de objetos.                         | <i>id, originalName, key, mimeType, size, subjectId, uploadedBy, isPublic e deletedAt.</i> |
| <i>classrooms</i>         | Representa as turmas criadas pelos professores.                                                   | <i>id, name, description, teacherId, createdAt e updatedAt.</i>                            |
| <i>classroom_members</i>  | Representa os vínculos entre estudantes e turmas.                                                 | <i>id, classroomId, studentId, status e createdAt.</i>                                     |
| <i>classroom_subjects</i> | Representa a associação entre turmas e disciplinas.                                               | <i>id, classroomId, subjectId e createdAt.</i>                                             |
| <i>classroom_invites</i>  | Armazena os convites temporários utilizados para solicitar ingresso em turmas.                    | <i>id, classroomId, token, expiresAt e createdAt.</i>                                      |
| <i>assignments</i>        | Representa as atividades acadêmicas criadas pelos professores.                                    | <i>id, subjectId, teacherId, title, description, dueDate e dados do anexo.</i>             |

|                               |                                                                                                |                                                                                                             |
|-------------------------------|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <i>assignment_submissions</i> | Armazena os metadados das entregas realizadas pelos estudantes.                                | <i>id, assignmentId, studentId, key, originalName, mimeType, size, resubmitAllowed e submittedAt.</i>       |
| <i>reports</i>                | Armazena as denúncias registradas pelos usuários.                                              | <i>id, reporterId, targetType, targetUserId, targetFileId, reason, status, resolvedBy e resolutionNote.</i> |
| <i>storage_cleanup_jobs</i>   | Mantém registros de objetos cuja remoção do armazenamento necessita de processamento posterior | <i>id, key, attempts, lastError, createdAt, updatedAt</i>                                                   |

Fonte: autor (2026).

### 5.7.2 Relacionamentos e cardinalidades

O usuário constitui uma das entidades centrais do modelo. Dependendo do papel associado à conta, ele pode exercer diferentes relações com as demais entidades.

Um professor pode criar várias disciplinas, enquanto cada disciplina pertence a um único professor. De maneira semelhante, um professor pode criar várias turmas e atividades, mas cada turma e atividade possui apenas um professor responsável.

A relação entre turmas e estudantes é do tipo muitos para muitos, representada pela entidade associativa *classroom\_members*. Essa estrutura permite que um estudante participe de várias turmas e que uma turma possua vários estudantes. O atributo status diferencia membros ativos de solicitações ainda pendentes.

A associação entre turmas e disciplinas também é do tipo muitos para muitos e é representada pela entidade *classroom\_subjects*. Uma turma pode reunir várias disciplinas, e uma mesma disciplina pode ser vinculada a diferentes turmas do professor responsável.

Uma turma pode possuir vários convites, enquanto cada convite pertence a uma única turma. Os convites são identificados por *tokens* únicos e possuem data de expiração.

Cada disciplina pode possuir vários materiais e várias atividades. Os materiais são representados pela entidade *files*, que mantém os metadados do objeto armazenado externamente. As atividades são representadas pela entidade *assignments*.

Uma atividade pode possuir várias entregas, mas cada estudante pode realizar somente uma entrega por atividade. Essa restrição é representada pela unicidade da combinação entre *assignmentId* e *studentId* na entidade *assignment\_submissions*.

Um usuário pode registrar várias denúncias. Cada denúncia possui um único denunciante e pode ter como alvo um usuário ou um arquivo. Quando a denúncia atinge uma situação final, também pode ser associada ao administrador responsável pela decisão.

A entidade *storage\_cleanup\_jobs* possui finalidade operacional e não mantém relacionamento obrigatório com as demais entidades do modelo. Seus registros identificam objetos pendentes de remoção por meio da chave utilizada no serviço de armazenamento, permitindo que a tentativa de limpeza permaneça registrada mesmo quando o registro de domínio que originou a operação já não estiver presente no banco de dados.

Os relacionamentos descritos demonstram como as entidades da aplicação se integram para representar as regras de negócio do RepolF, assegurando a consistência dos dados e o controle das interações entre usuários, disciplinas, turmas, materiais, atividades e denúncias.

### 5.7.3 Integridade e restrições do modelo

O modelo de dados do RepolF utiliza restrições de integridade para assegurar a consistência das informações armazenadas. As chaves primárias identificam unicamente cada registro, enquanto as chaves estrangeiras preservam os

relacionamentos entre as entidades.

Restrições de unicidade são aplicadas aos atributos que não podem possuir valores duplicados, como nome de usuário, endereço de e-mail e tokens utilizados em processos de autenticação e convites. Também são empregadas nas entidades associativas para impedir registros duplicados, como nos vínculos entre estudantes e turmas, nas associações entre turmas e disciplinas e nas entregas de atividades.

A aplicação adota exclusão lógica para as entidades *user* e *files*, por meio do atributo *deletedAt*. Nesses casos, o registro permanece armazenado no banco de dados, permitindo sua restauração quando necessário, mas deixa de ser considerado nas operações convencionais da aplicação.

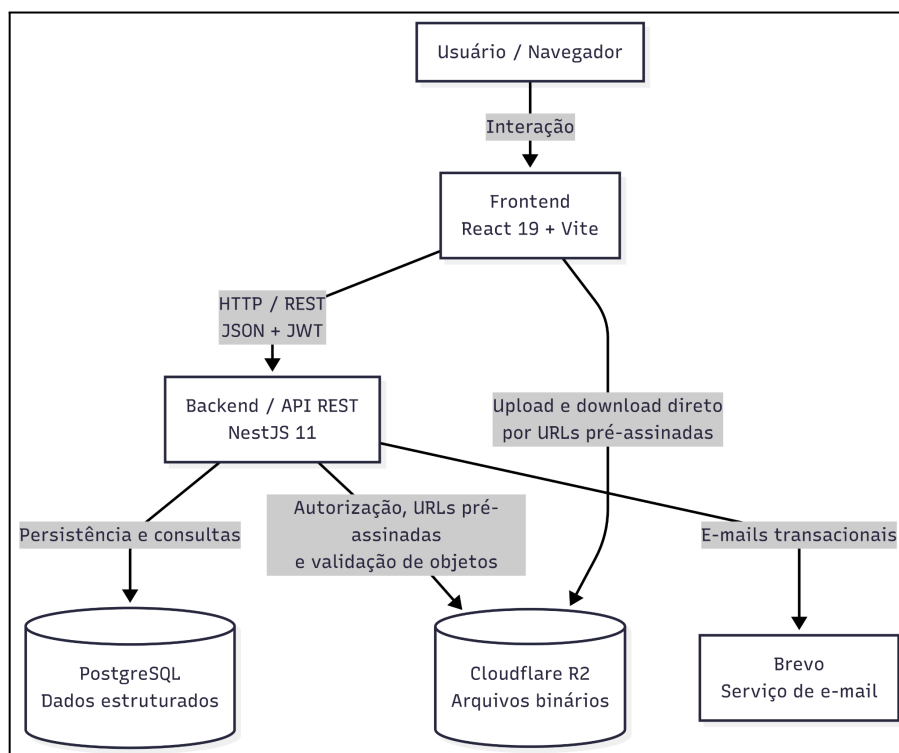
Quando uma operação envolve a remoção definitiva de objetos do armazenamento, a aplicação promove sua exclusão no Cloudflare R2. Nas situações contempladas pelo mecanismo de limpeza em que essa remoção não possa ser concluída imediatamente, a chave do objeto permanece registrada para nova tentativa, reduzindo a possibilidade de objetos órfãos e contribuindo para a consistência entre os metadados persistidos e o armazenamento de objetos.

## 6 ARQUITETURA E TECNOLOGIAS

A arquitetura do RepolF foi projetada para separar as responsabilidades entre apresentação, processamento das regras de negócio, persistência de dados e armazenamento de arquivos. A aplicação adota o modelo cliente-servidor, composto por um *frontend* executado no navegador e um *backend* responsável pela disponibilização da *API REST*, autenticação, autorização e integração com os serviços externos.

Os dados estruturados são armazenados em banco de dados PostgreSQL, enquanto os arquivos binários são mantidos separadamente no *Cloudflare R2*. O sistema também utiliza um serviço externo de e-mail para operações relacionadas à verificação de contas, recuperação de senha e notificações. A Figura 6 apresenta a arquitetura geral do RepolF.

Figura 6 - Arquitetura geral do RepolF



Fonte: autor (2026).

### 6.1 ARQUITETURA GERAL DA APLICAÇÃO

O RepolF possui dois componentes principais de *software*: o *frontend*, responsável pela apresentação das interfaces e pela interação com os usuários, e o *backend*, responsável pelas regras de negócio, autenticação, autorização e integração com os mecanismos de persistência e serviços externos.

A comunicação entre *frontend* e *backend* ocorre por meio de requisições *HTTP* à *API REST*. Nas funcionalidades protegidas, as requisições incluem as informações necessárias à identificação da sessão, permitindo que o *backend* determine se o usuário possui permissão para executar determinada operação.

O *backend* também é responsável pela comunicação com o PostgreSQL, com o *Cloudflare R2* e com o serviço de e-mail. Dessa forma, credenciais e regras sensíveis permanecem concentradas no servidor, enquanto o *frontend* atua principalmente na interação com o usuário e na apresentação das informações.

Nas operações envolvendo arquivos, a *API* autoriza inicialmente a ação e fornece ao *frontend* uma *URL* temporária. A transferência pode então ocorrer diretamente entre o navegador e o serviço de armazenamento, evitando que o conteúdo integral dos arquivos precise trafegar pelo *backend*.

## 6.2 FRONTEND E BACKEND

O *frontend* foi desenvolvido utilizando *React 19* e *TypeScript*. O *React* estrutura aplicações a partir de componentes responsáveis por partes da interface e por seus respectivos comportamentos (React, [s. d.]). No RepolF, essa abordagem foi utilizada para organizar as responsabilidades da interface entre componentes de apresentação, navegação, comunicação com a API e estilização. Essa organização, favorecida pelo ecossistema do *React*, utiliza o *Vite* como ferramenta de desenvolvimento e construção da aplicação (Vite, [s. d.]), o *Tailwind Cascading Style Sheets (CSS)* para estilização das interfaces, o *React Router Document Object Model (DOM)* para o gerenciamento das rotas e o *Axios* para a comunicação com a *API REST*.

A comunicação com a *API* é realizada por meio do *Axios*, utilizando uma configuração centralizada para as requisições *HTTP*. O *frontend* também possui mecanismos responsáveis por proteger páginas e diferenciar funcionalidades

conforme o perfil autenticado.

O *backend* foi desenvolvido utilizando *NestJS 11* e *TypeScript*. O *framework* fornece uma estrutura para construção de aplicações no ecossistema *Node.js* e organiza recursos relacionados a *controllers*, *providers* e módulos (NestJS, [s. d.]). No RepolF, sua arquitetura modular distribui as responsabilidades entre os domínios de autenticação, usuários, disciplinas, arquivos, turmas, atividades, denúncias, administração, armazenamento e comunicação por e-mail.

Os *controllers* recebem as requisições realizadas à *API*, enquanto os *services* concentram as principais regras de negócio. Os dados recebidos são validados antes de seu processamento, contribuindo para impedir informações inválidas ou incompatíveis com as operações solicitadas.

Essa separação permite manter a interface independente da lógica de negócio e da camada de persistência, favorecendo a organização e a manutenção da aplicação.

### 6.3 PERSISTÊNCIA E ARMAZENAMENTO DE ARQUIVOS

O RepolF utiliza o PostgreSQL como sistema de gerenciamento do banco de dados relacional responsável pela persistência das informações estruturadas da aplicação (PostgreSQL Global Development Group, 2026). A integração entre o backend e o banco é realizada por meio do *TypeORM*, ferramenta de mapeamento objeto-relacional com suporte ao PostgreSQL e a recursos como entidades, relacionamentos, índices, transações e migrações (TypeORM, [s. d.]).

O banco armazena informações relacionadas a usuários, disciplinas, turmas, atividades, entregas, denúncias e metadados dos materiais, enquanto os arquivos propriamente ditos são mantidos no *Cloudflare R2*.

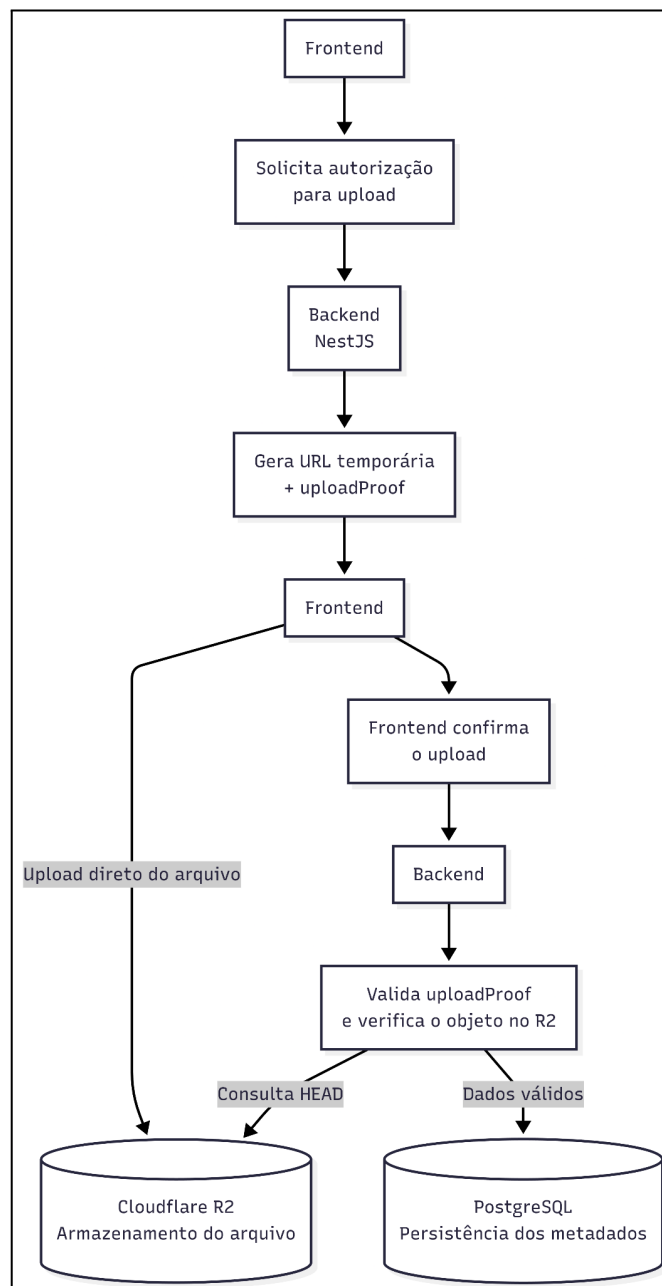
Essa separação evita o armazenamento dos arquivos binários diretamente no banco relacional e permite que cada tecnologia seja utilizada de acordo com sua finalidade.

Nos *uploads*, o *backend* verifica inicialmente se o usuário possui autorização para realizar a operação. Após essa validação, é disponibilizada uma *URL*

temporária que permite ao *frontend* enviar o arquivo diretamente ao *R2*. O *Cloudflare R2* permite o uso de *URLs* pré-assinadas para conceder acesso temporário a operações específicas sobre objetos, incluindo envio, consulta de metadados e recuperação de arquivos, sem exposição das credenciais utilizadas pelo serviço (Cloudflare, 2026).

A Figura 7 resume esse processo.

Figura 7 - Fluxo de upload de arquivos no RepolF



Fonte: autor (2026).

Entre os mecanismos utilizados durante essa confirmação encontra-se uma prova temporária associada à operação de *upload*. Esse recurso atua como uma verificação adicional de segurança, mas integra o fluxo de armazenamento e não constitui um componente independente da arquitetura.

Além do envio, a aplicação possui mecanismos para manter a consistência entre o banco de dados e o armazenamento de objetos. Caso a remoção de determinado arquivo no *R2* não possa ser concluída imediatamente, sua chave pode ser registrada em *storage\_cleanup\_jobs*, permitindo que novas tentativas sejam realizadas posteriormente.

Essa estratégia é utilizada em operações que podem envolver exclusão ou substituição de arquivos, reduzindo a possibilidade de objetos permanecerem no armazenamento sem necessidade.

## 6.4 SEGURANÇA E SERVIÇOS EXTERNOS

A autenticação do RepolF utiliza *JSON Web Token (JWT)*. As senhas são armazenadas utilizando *hash* com *bcrypt*, enquanto o acesso às funcionalidades protegidas é controlado por *guards* no *backend*. A autorização considera os papéis de estudante, professor e administrador.

Além do papel, determinadas operações verificam a propriedade do recurso ou o vínculo acadêmico do usuário, como ocorre no gerenciamento de disciplinas, turmas, atividades e materiais.

A aplicação emprega mecanismos complementares de segurança, incluindo validação das entradas recebidas, limitação da frequência de requisições, configuração de cabeçalhos *HTTP*, política de *Cross-Origin Resource Sharing (CORS)*, restrições ao envio de arquivos e utilização de *URLs* temporárias para operações de *upload* e *download*.

Para o envio de mensagens, o RepolF utiliza o *Brevo* como serviço externo de e-mail. As mensagens são utilizadas principalmente nos processos de verificação de conta, recuperação de senha e notificações associadas a determinadas funcionalidades acadêmicas.

## 6.5 TECNOLOGIAS UTILIZADAS

O Quadro 14 apresenta as principais tecnologias utilizadas na implementação do RepolF.

Quadro 14 - Principais tecnologias utilizadas no RepolF

| Camada/Finalidade         | Tecnologia                            |
|---------------------------|---------------------------------------|
| Interface                 | <i>React 19</i>                       |
| Linguagem                 | <i>TypeScript</i>                     |
| <i>Build frontend</i>     | <i>Vite</i>                           |
| Estilização               | <i>Tailwind CSS</i>                   |
| Roteamento                | <i>React Router DOM</i>               |
| Comunicação <i>HTTP</i>   | <i>Axios</i>                          |
| <i>Backend/API</i>        | <i>NestJS 11</i>                      |
| <i>ORM</i>                | <i>TypeORM</i>                        |
| Banco de dados            | PostgreSQL                            |
| Armazenamento de arquivos | <i>Cloudflare R2</i>                  |
| Autenticação              | <i>JWT</i>                            |
| <i>Hash</i> de senhas     | <i>bcrypt</i>                         |
| Segurança <i>HTTP</i>     | <i>Helmet</i>                         |
| Limitação de requisições  | <i>NestJS Throttler</i>               |
| E-mail                    | <i>Brevo</i>                          |
| Testes do <i>backend</i>  | <i>Jest e Supertest</i>               |
| Testes do <i>frontend</i> | <i>Vitest e React Testing Library</i> |

Fonte: autor (2026).

A combinação dessas tecnologias permitiu separar as responsabilidades da plataforma entre interface, regras de negócio, persistência relacional, armazenamento de arquivos e serviços externos. A arquitetura adotada também

possibilitou que operações relacionadas a arquivos fossem realizadas sem concentrar sua transferência no *backend*, mantendo neste componente as decisões relacionadas à autorização, validação e integridade das operações.

## 7 IMPLEMENTAÇÃO E FUNCIONALIDADES DO REPOIF

Este capítulo apresenta as principais funcionalidades implementadas no RepoiF e a forma como elas foram disponibilizadas aos diferentes perfis de usuário. A descrição concentra-se no comportamento efetivamente presente na versão final da aplicação, complementando os requisitos e regras de negócio apresentados no Capítulo 5 e a arquitetura descrita no Capítulo 6.

A implementação foi organizada de acordo com os principais domínios da plataforma, abrangendo autenticação e gerenciamento de contas, perfis, disciplinas e materiais, turmas, atividades e entregas, além das funcionalidades administrativas e de moderação.

### 7.1 AUTENTICAÇÃO E GERENCIAMENTO DE CONTAS

O RepoiF disponibiliza um conjunto de funcionalidades destinadas ao controle de acesso à plataforma. O cadastro público permite a criação de contas com os perfis de estudante ou professor, sendo o papel de administrador restrito ao módulo administrativo.

Durante o cadastro, são informados nome de usuário, endereço de e-mail, senha e perfil. Após a criação da conta, o sistema encaminha uma mensagem para confirmação do endereço eletrônico. Enquanto essa verificação não é concluída, o usuário não pode realizar a autenticação.

Também foram implementados mecanismos para reenvio do e-mail de verificação e recuperação de senha. No processo de recuperação, o usuário informa seu endereço de e-mail e, caso exista uma conta correspondente, recebe um *link* temporário para definição de uma nova senha.

Após a autenticação, cada usuário é direcionado ao ambiente correspondente ao seu papel na plataforma: professores acessam o painel de disciplinas, estudantes são encaminhados ao próprio perfil e administradores ao painel administrativo.

A Figura 8 apresenta as principais interfaces relacionadas ao acesso à plataforma.

Figura 8 - Interfaces de autenticação e gerenciamento de conta do RepolF

The figure displays three distinct user interface screens for the RepolF platform, all featuring the RepolF logo and the tagline 'Organize. Compartilhe. Estude.' at the top.

- Left Screen (Login):** Titled 'Entrar na conta', it prompts users to 'Acesse com username ou email'. It includes input fields for 'Username ou email' (with the example 'Ex: joaosilva ou joao@email.com') and 'Senha' (with a masked password '\*\*\*\*\*' and a toggle for visibility). A blue 'Entrar' button is at the bottom. Links for 'Esqueci minha senha' and 'Não tem conta? Cadastre-se' are present, along with a 'Reenviar' link for users who haven't received a verification email.
- Middle Screen (Registration):** Titled 'Criar conta', it asks users to 'Escolha seu perfil e preencha os dados'. It offers two roles: 'Aluno' (selected) and 'Professor'. A note states: 'Acesse conteúdos de professores e faça download dos arquivos disponibilizados.' Below are fields for 'Username' (example: 'Ex: joaosilva (3-144 caracteres)'), 'Email' (example: 'Ex: joao@email.com'), and 'Senha' (with a 'Mínimo 8 caracteres' requirement and a toggle for visibility). A blue 'Criar conta' button is at the bottom, with a link 'Já tem conta? Entrar' below it.
- Right Screen (Forgot Password):** Titled 'Esqueceu a senha?', it prompts users to 'Informe seu email e enviaremos um link de redefinição'. It features an 'Email' input field (example: 'seu@email.com') and a blue 'Enviar link de redefinição' button. A link 'Lembrou a senha? Entrar' is at the bottom.

Fonte: autor (2026).

Além dos processos de entrada e recuperação de conta, usuários autenticados podem alterar seu nome de exibição e modificar a própria senha. O nome de usuário utilizado como identificador do perfil permanece inalterável pela interface de edição.

## 7.2 PERFIS E CONSULTA DE PROFESSORES

Cada usuário possui um perfil que apresenta suas informações básicas e, quando se trata do próprio perfil, indicadores relacionados à sua utilização da plataforma.

Para professores, são apresentados dados relacionados à quantidade de disciplinas, turmas, materiais e atividades cadastradas. Para estudantes, o perfil apresenta informações como número de turmas, entregas realizadas e atividades ainda pendentes.

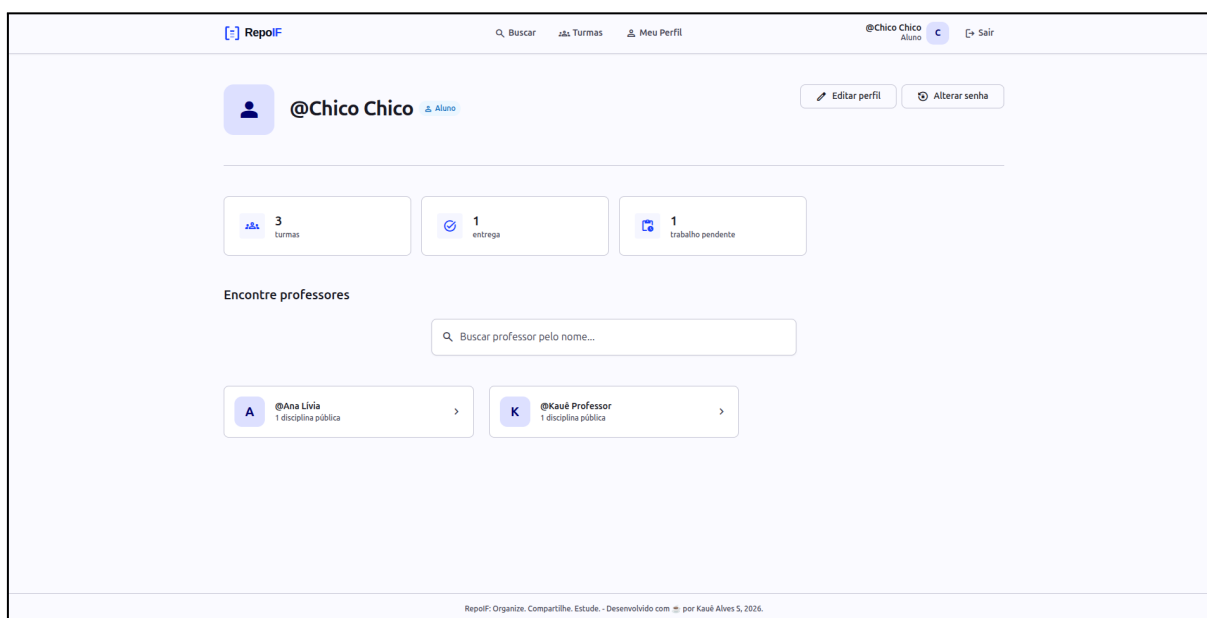
O professor também pode utilizar seu perfil como ponto de acesso às disciplinas que mantém na plataforma. Quando outro usuário consulta esse perfil, a

relação de disciplinas apresentada depende das permissões existentes.

No caso dos estudantes, a consulta de professores foi implementada considerando os vínculos acadêmicos existentes. Dessa forma, o estudante visualiza e pesquisa professores com os quais possui vínculo por meio de uma turma ativa. Quando acessa o perfil de um desses professores, pode visualizar as disciplinas públicas e as disciplinas privadas às quais possui acesso por meio das turmas em que participa.

A aplicação também disponibiliza uma interface de busca de professores, utilizando pesquisa textual e paginação.

Figura 9 - Perfil de usuário e consulta de professores no RepolF



Fonte: autor (2026).

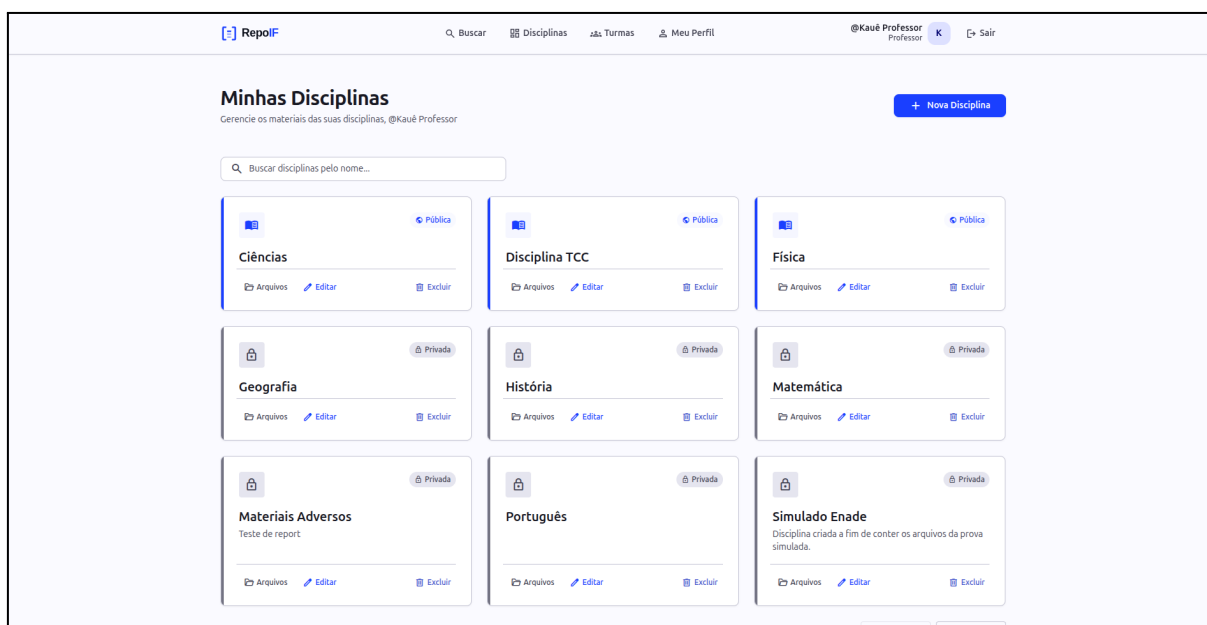
Os próprios perfis também disponibilizam ações relacionadas à edição dos dados pessoais e à alteração da senha. Quando o perfil pertence a outro usuário, a interface permite registrar uma denúncia quando necessário.

### 7.3 DISCIPLINAS E GERENCIAMENTO DE MATERIAIS

O gerenciamento de disciplinas constitui uma das funcionalidades centrais do RepolF. Professores podem criar, editar e excluir suas próprias disciplinas, definindo nome, descrição e visibilidade.

O painel principal do professor apresenta suas disciplinas em formato de cartões, permitindo realizar buscas pelo nome e acessar diretamente seus respectivos conteúdos. A diferenciação entre disciplinas públicas e privadas é apresentada visualmente na interface.

Figura 10 - Painel de disciplinas do professor



Fonte: autor (2026).

Ao acessar uma disciplina, o professor pode adicionar e gerenciar materiais didáticos. O envio é realizado a partir da seleção do arquivo e da definição de sua visibilidade. Conforme apresentado no capítulo anterior, os arquivos são armazenados no *Cloudflare R2* e seus metadados permanecem registrados no PostgreSQL.

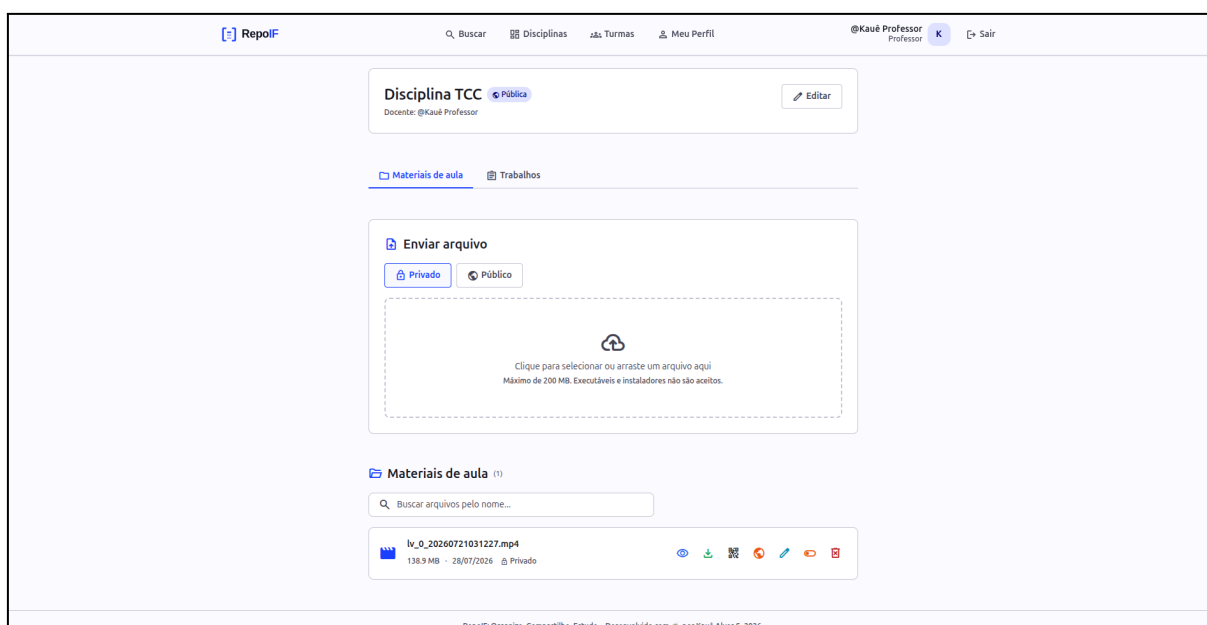
A interface da disciplina permite:

- pesquisar materiais pelo nome;
- visualizar e baixar arquivos;
- renomear materiais;
- alterar sua visibilidade;
- desabilitar e reabilitar arquivos;
- excluir materiais definitivamente;
- compartilhar materiais por *QR Code* quando permitido.

A visibilidade do material é independente da visibilidade da disciplina. Um arquivo definido como público pode ser acessado por meio do respectivo *link* mesmo sem autenticação. Por outro lado, um arquivo privado permanece restrito ao professor proprietário e não se torna automaticamente acessível apenas pelo fato de um estudante participar de uma turma associada à disciplina.

Quando um material é desabilitado, ele deixa de aparecer para os demais usuários, mas continua disponível ao professor proprietário, que pode reabilitá-lo posteriormente. A exclusão definitiva remove permanentemente o registro e promove a remoção do respectivo objeto armazenado, utilizando o mecanismo de limpeza posterior nos fluxos em que ele for necessário.

Figura 11 - Gerenciamento de materiais de uma disciplina



Fonte: autor (2026).

O professor também pode gerar QR Code para compartilhamento de materiais. No caso de arquivos privados, a interface solicita confirmação antes da geração do código, uma vez que o QR Code utiliza uma URL temporária pré-assinada do serviço de armazenamento, possibilitando o acesso ao arquivo durante seu período de validade.

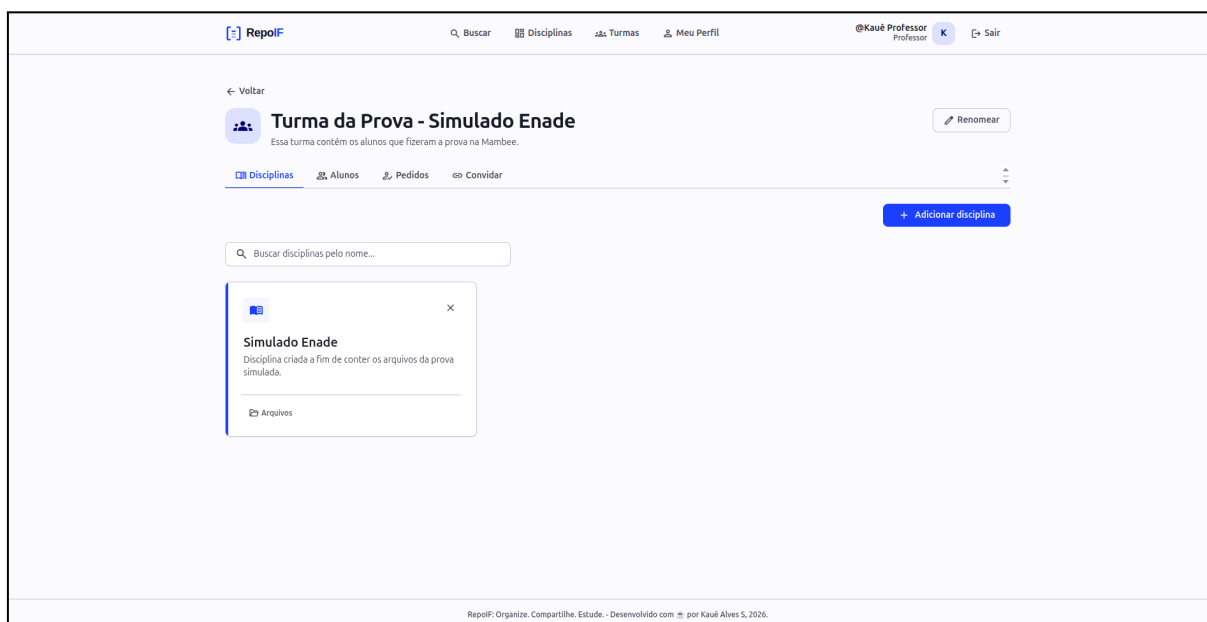
## 7.4 TURMAS, DISCIPLINAS E INGRESSO DE ESTUDANTES

O módulo de turmas foi implementado para representar a organização dos estudantes e das disciplinas sob responsabilidade de um professor.

O professor pode criar turmas, renomeá-las e excluí-las. A página principal apresenta as turmas pertencentes ao docente, enquanto estudantes visualizam somente aquelas em que possuem participação ativa.

Dentro de uma turma, o professor pode associar disciplinas já existentes ou criar uma nova disciplina durante o próprio processo de associação. Quando uma nova disciplina é criada a partir desse fluxo, ela é inicialmente configurada como privada.

Figura 12 - Gerenciamento de turma no RepolF



Fonte: autor (2026).

O ingresso de estudantes em uma turma também pode ocorrer por meio de convite. Nesse processo, o professor gera um *link* temporário, que pode ser disponibilizado diretamente ou por meio de *QR Code*. Ao acessar o convite, o estudante encaminha uma solicitação de ingresso à turma. A solicitação permanece pendente até ser analisada pelo professor responsável, que pode aceitá-la ou rejeitá-la. Quando aceita, o estudante passa a integrar a turma como membro ativo.

O sistema também envia notificações por e-mail em determinados eventos desse processo, como solicitações de ingresso, adição direta de estudantes e aceite

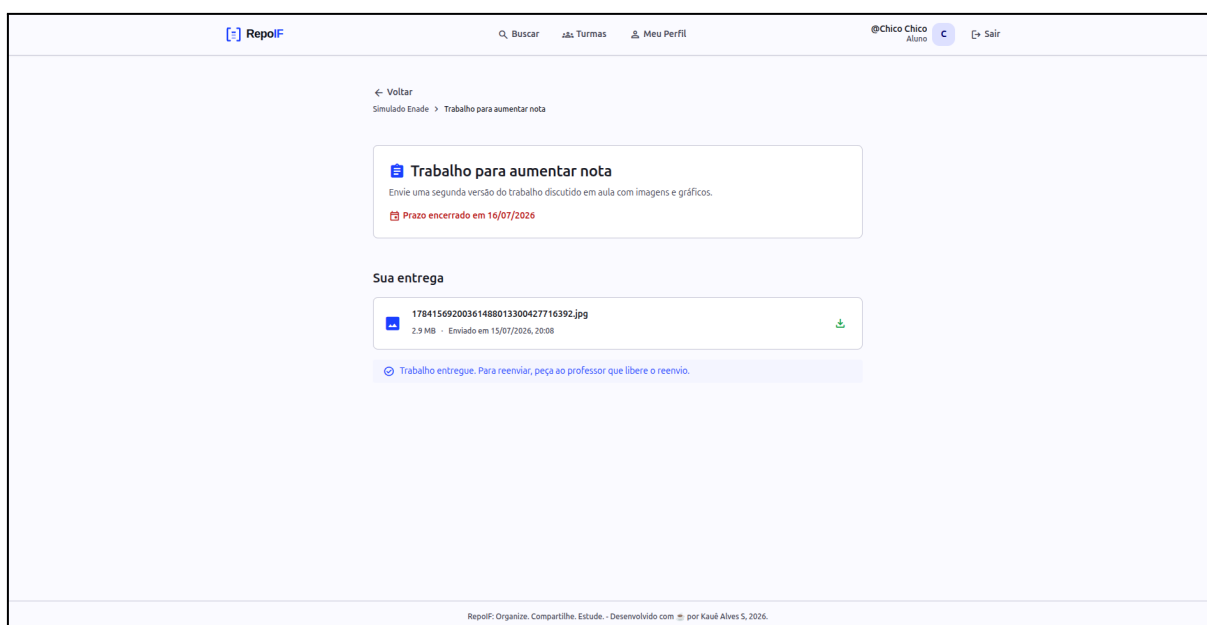
de pedidos.

## 7.5 ATIVIDADES E ENTREGAS

As atividades são criadas pelos professores dentro das disciplinas sob sua responsabilidade. Cada atividade possui título, prazo de entrega, descrição opcional e pode conter um arquivo anexo.

A criação e a consulta das atividades estão integradas à página da disciplina, possibilitando ao professor acompanhar os conteúdos e trabalhos relacionados ao mesmo contexto acadêmico.

Figura 13 - Gerenciamento de atividades no RepolF



Fonte: autor (2026).

Os estudantes com acesso à disciplina podem consultar as atividades disponíveis. Entretanto, a possibilidade de visualizar uma atividade não significa necessariamente que o usuário poderá realizar uma entrega. Para efetuar a submissão, o estudante deve possuir vínculo ativo com uma turma associada à disciplina e o prazo da atividade deve permitir a operação.

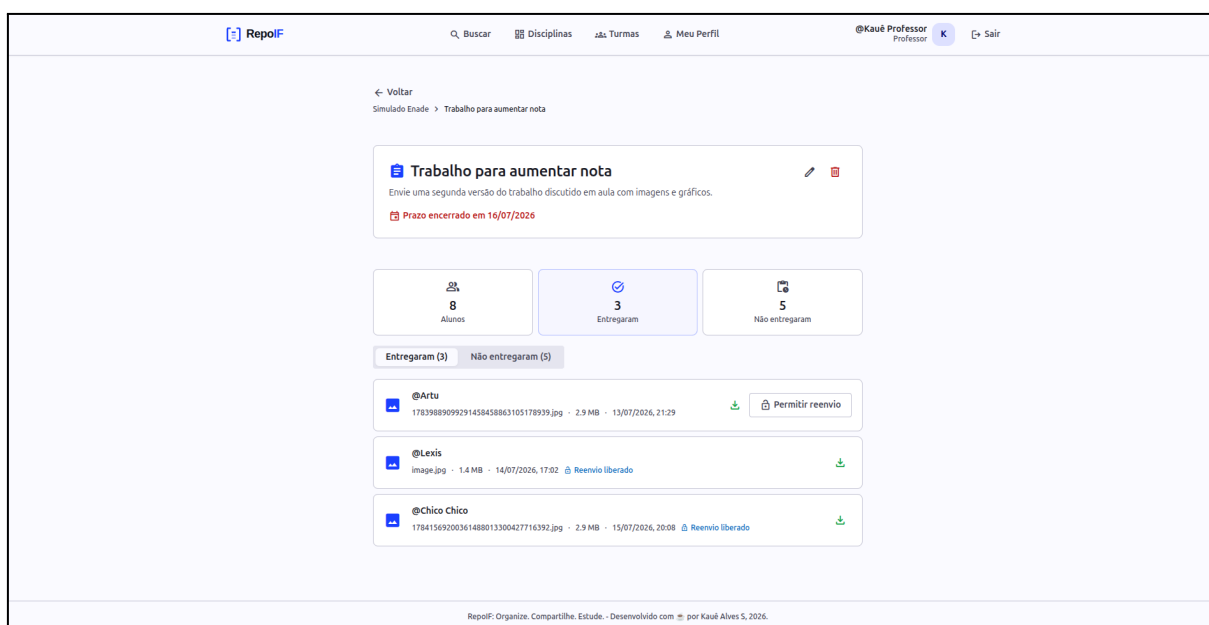
O estudante realiza a entrega por meio do envio de um arquivo. Após a submissão, pode consultar e baixar o próprio arquivo entregue. O sistema mantém

no máximo uma entrega por estudante para cada atividade.

Quando necessário, o professor pode autorizar individualmente o reenvio. Nesse caso, o estudante recebe novamente a possibilidade de realizar a submissão e o novo arquivo substitui a entrega anterior. Após o reenvio, a autorização é automaticamente desabilitada.

A Figura 14 apresenta a interface utilizada para acompanhamento de uma atividade.

Figura 14 - Detalhes de atividade e acompanhamento de entregas



Fonte: autor (2026).

Na visão do professor, a página apresenta informações sobre as entregas realizadas e os estudantes que ainda não enviaram a atividade. O docente pode baixar os arquivos recebidos e liberar reenvios individualmente.

Essa funcionalidade permite ao RepolIF atuar não somente como espaço para armazenamento de materiais, mas também como ambiente de organização dos conteúdos e atividades vinculados às disciplinas e turmas.

## 7.6 DENÚNCIAS E MODERAÇÃO

Usuários autenticados podem registrar denúncias relacionadas a outros

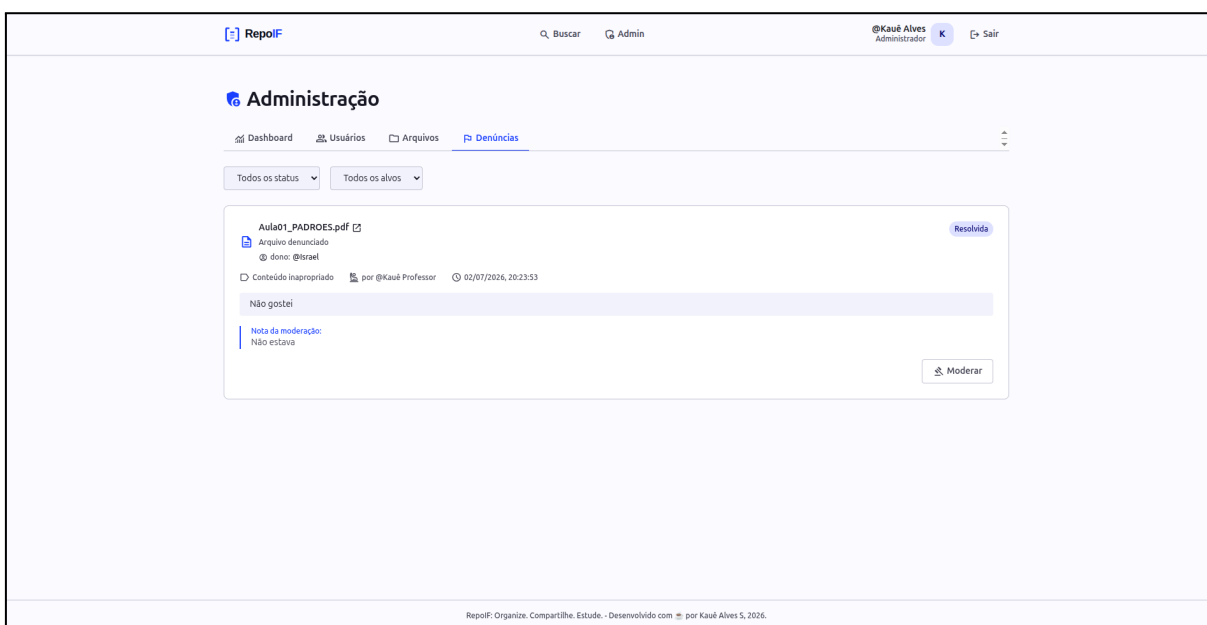
usuários ou a materiais disponibilizados na plataforma. Durante o registro, é selecionado um motivo e pode ser adicionada uma descrição complementar.

O sistema impede que um usuário denuncie a própria conta e evita a existência simultânea de denúncias pendentes duplicadas para o mesmo alvo.

As denúncias são encaminhadas ao ambiente administrativo, no qual podem ser consultadas e analisadas. O administrador dispõe de filtros por situação e tipo de alvo e pode alterar o estado da denúncia, além de registrar uma nota referente à decisão tomada.

Quando a denúncia estiver relacionada a um arquivo, o administrador pode acessar o material para auxiliar no processo de moderação.

Figura 15 - Interface de moderação de denúncias



Fonte: autor (2026).

Essa funcionalidade complementa os mecanismos de controle da plataforma, permitindo que conteúdos ou comportamentos inadequados sejam informados pelos próprios usuários e posteriormente analisados pela administração.

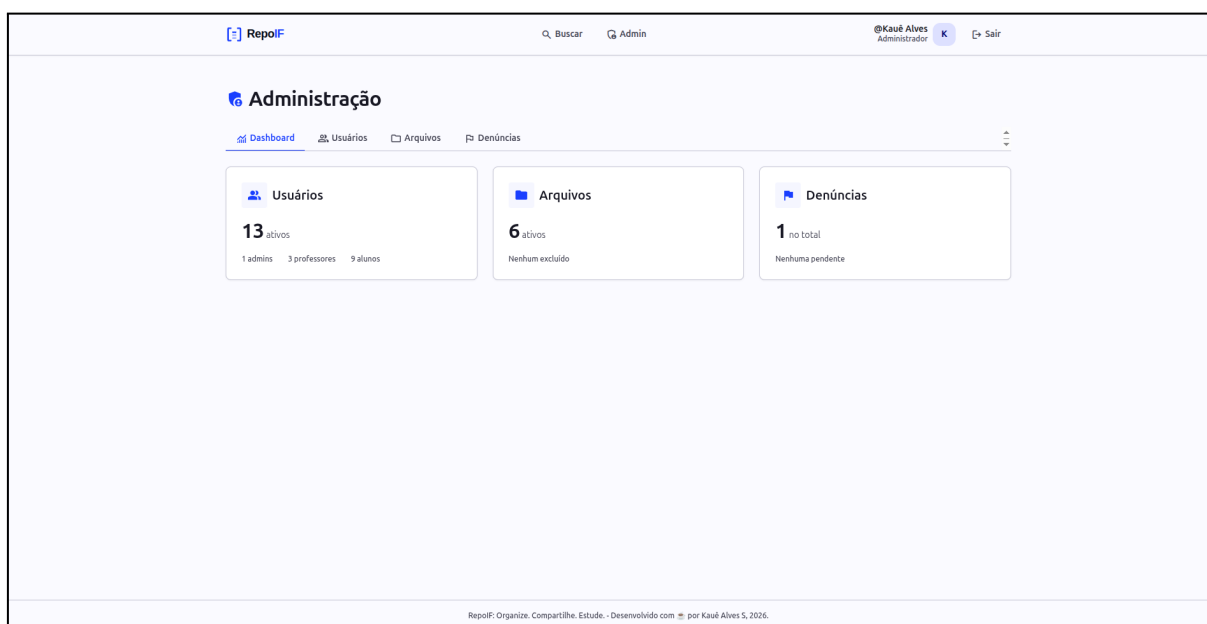
## 7.7 FUNCIONALIDADES ADMINISTRATIVAS

O papel de administrador possui um ambiente próprio, separado das

funcionalidades acadêmicas de professores e estudantes.

O painel administrativo apresenta indicadores gerais relacionados aos usuários, arquivos e denúncias registrados no sistema. A partir desse painel, o administrador pode acessar os módulos específicos de gerenciamento.

Figura 16 - Painel administrativo do RepoIF



Fonte: autor (2026).

No gerenciamento de usuários, são disponibilizadas funções de busca, paginação e filtragem por papel. O administrador pode criar contas previamente verificadas, modificar o papel de outros usuários, realizar exclusão lógica e restaurar contas anteriormente excluídas.

Como medida de proteção administrativa, o sistema impede que um administrador altere o próprio papel ou exclua sua própria conta por meio dessas funcionalidades.

No módulo de arquivos, o administrador pode consultar materiais cadastrados na plataforma, realizar buscas e incluir registros excluídos logicamente na visualização. Também pode baixar arquivos para fins de moderação, realizar exclusão lógica, restaurar registros ou executar uma exclusão definitiva.

A área administrativa de denúncias permite consultar ocorrências, aplicar

filtros e registrar a resolução adotada.

Essas funcionalidades foram implementadas com o objetivo de fornecer mecanismos de manutenção e moderação sem conceder essas operações aos perfis acadêmicos comuns.

## 7.8 SÍNTESE DAS FUNCIONALIDADES IMPLEMENTADAS

A versão desenvolvida do RepolF contempla os principais fluxos definidos durante a especificação da solução. Professores podem estruturar disciplinas, disponibilizar materiais, organizar turmas e criar atividades; estudantes podem participar de turmas, consultar conteúdos e realizar entregas; e administradores possuem recursos específicos para gerenciamento e moderação.

As funcionalidades implementadas mantêm separação entre os diferentes perfis, aplicando regras de autorização tanto na interface quanto no *backend*. Da mesma forma, operações envolvendo arquivos, exclusões e submissões foram integradas aos mecanismos de persistência e armazenamento descritos no Capítulo 6.

A implementação resultante foi posteriormente submetida aos procedimentos de validação técnica e avaliação de usabilidade definidos na metodologia. Os resultados dessas etapas são apresentados no capítulo seguinte.

## 8 RESULTADOS E VALIDAÇÃO

Este capítulo apresenta os resultados obtidos com o desenvolvimento e a validação do RepolF. A análise foi estruturada em duas perspectivas complementares. A primeira corresponde à validação técnica da aplicação, realizada por meio de testes automatizados no *backend* e no *frontend* e de testes de integração. A segunda corresponde à avaliação da usabilidade das interfaces com base nas heurísticas de Nielsen (1994).

Os resultados apresentados concentram-se na verificação do funcionamento técnico da solução e no atendimento aos requisitos estabelecidos. Não se pretende, portanto, mensurar impactos sobre o processo de ensino e aprendizagem ou a satisfação da comunidade acadêmica, uma vez que não foram realizados experimentos ou levantamentos com estudantes e professores.

### 8.1 RESULTADOS DA IMPLEMENTAÇÃO

O desenvolvimento resultou em uma aplicação *web* funcional composta por interfaces e funcionalidades destinadas aos perfis de estudante, professor e administrador. A implementação contemplou os principais requisitos especificados para os perfis de professor, estudante e administrador, abrangendo gerenciamento acadêmico, compartilhamento de materiais, controle de acesso, atividades, entregas e funcionalidades administrativas.

A implementação dessas funcionalidades demonstra que os componentes necessários ao escopo definido para o RepolF foram desenvolvidos. Entretanto, a presença das funcionalidades não foi considerada, isoladamente, evidência suficiente de seu correto funcionamento. Por esse motivo, a aplicação também foi submetida aos procedimentos de validação técnica apresentados nas seções seguintes.

### 8.2 VALIDAÇÃO TÉCNICA

A validação técnica do RepolF foi realizada por meio de testes automatizados destinados a verificar regras de negócio, componentes da interface, mecanismos de

autenticação e autorização, validações, tratamento de erros e operações relacionadas à persistência dos dados.

Foram utilizadas ferramentas específicas para as diferentes camadas da aplicação. No *backend*, os testes foram implementados principalmente com *Jest*, utilizando também *Supertest* nos cenários envolvendo requisições *HTTP*. No *frontend*, foram empregados *Vitest* e *React Testing Library*.

Além dos fluxos considerados válidos, foram incluídos cenários envolvendo dados incorretos, operações não autorizadas, duplicidades e demais situações que poderiam comprometer o comportamento esperado da aplicação.

Considerando todas as suítes e seus respectivos ambientes de execução, foram executados 418 testes distintos no *backend* e 159 no *frontend*, totalizando 577 testes automatizados aprovados.

### 8.2.1 Resultados dos testes do backend

A suíte do *backend* foi composta por 36 arquivos de teste e 418 casos de teste distintos. Parte desses casos pôde ser executada independentemente de uma instância real do banco de dados, enquanto outra parte foi destinada especificamente à validação da integração com PostgreSQL.

Na execução convencional do *backend*, foram aprovados 351 testes, distribuídos em 29 suítes executadas. Outros 67 casos foram intencionalmente ignorados nessa configuração por dependerem do ambiente de integração.

Quadro 15 - Síntese dos testes automatizados do *backend*

| Aspecto                                   | Resultado |
|-------------------------------------------|-----------|
| Suítes existentes                         | 36        |
| Suítes executadas e aprovadas             | 29        |
| Suítes dependentes de ambiente específico | 7         |
| Testes aprovados                          | 351       |
| Testes dependentes de ambiente específico | 67        |

|        |   |
|--------|---|
| Falhas | 0 |
|--------|---|

Fonte: autor (2026).

Os testes abrangeram os principais módulos da aplicação, incluindo autenticação, gerenciamento de usuários, disciplinas, materiais, turmas, atividades, entregas, denúncias, funcionalidades administrativas e mecanismos associados ao armazenamento de arquivos.

Nos módulos de autenticação e autorização, foram verificadas situações como *login*, validação dos mecanismos de acesso, diferenciação entre papéis e bloqueio de operações não autorizadas. Nos módulos acadêmicos, foram avaliadas regras relacionadas a disciplinas, turmas, membros, convites, atividades e submissões.

Também foram incluídos testes referentes à validação dos arquivos enviados, controle das operações de armazenamento, tratamento de registros duplicados e mecanismos utilizados para preservar a consistência das informações.

Os 67 casos dependentes de infraestrutura adicional foram posteriormente executados em seus respectivos ambientes, conforme apresentado na Seção 8.3. Dessa forma, todos os 418 testes distintos existentes no *backend* foram executados e aprovados durante o processo de validação.

### 8.2.2 Resultados dos testes do frontend

A validação automatizada do *frontend* foi realizada utilizando *Vitest* e *React Testing Library*. A suíte foi composta por 33 arquivos de teste, totalizando 159 casos executados e aprovados, sem ocorrência de falhas.

Quadro 16 - Síntese dos testes automatizados do *frontend*

| Aspecto           | Resultado |
|-------------------|-----------|
| Arquivos de teste | 33        |
| Testes executados | 159       |
| Testes aprovados  | 159       |

|                       |                       |
|-----------------------|-----------------------|
| Testes com falha      | 0                     |
| Ferramenta principal  | Vitest                |
| Testes de componentes | React Testing Library |

Fonte: autor (2026).

Os testes contemplaram páginas e componentes relacionados à autenticação, cadastro, recuperação e verificação de conta, perfis, disciplinas, turmas, atividades e funcionalidades administrativas.

Também foram avaliados mecanismos utilizados transversalmente pela interface, como proteção de rotas, modais, notificações, ações de confirmação e comunicação com a *API*.

Os cenários verificaram comportamentos como preenchimento e validação de formulários, apresentação de mensagens, redirecionamentos, disponibilidade de ações conforme o perfil autenticado e resposta da interface diante de diferentes resultados das operações.

A execução final apresentou 33 arquivos de teste aprovados e 159 testes aprovados. Esse resultado fornece evidências de que os comportamentos contemplados pela suíte responderam conforme o esperado na versão avaliada.

### 8.2.3 Síntese dos testes automatizados

Considerando o *backend* e o *frontend*, o RepolF apresentou 577 casos de teste distintos aprovados.

Quadro 17 - Síntese da validação automatizada do RepolF

| Camada   | Testes distintos | Resultado |
|----------|------------------|-----------|
| Backend  | 418              | Aprovados |
| Frontend | 159              | Aprovados |
| Total    | 577              | Aprovados |

Fonte: autor (2026).

A quantidade apresentada corresponde aos casos de teste distintos existentes nas suítes, e não à soma do número de execuções realizadas pelos diferentes comandos. Alguns testes convencionais do *backend* também são executados durante a validação com PostgreSQL e, por esse motivo, não foram contabilizados novamente no total.

Os resultados demonstram que os cenários contemplados pelas suítes apresentaram o comportamento esperado durante a execução realizada para a versão avaliada. Entretanto, a aprovação dos testes não deve ser interpretada como garantia de ausência absoluta de defeitos, uma vez que a validação está limitada aos comportamentos e condições representados pelos casos implementados.

### 8.3 TESTES DE INTEGRAÇÃO

Além dos testes executados de forma isolada, foram realizados testes de integração utilizando uma instância PostgreSQL destinada especificamente ao ambiente de validação. O objetivo dessa etapa foi verificar comportamentos que dependem da interação efetiva entre a aplicação e o banco de dados, complementando os testes convencionais do *backend*.

Foram executados 59 testes relacionados à persistência e às regras de negócio dependentes do PostgreSQL, além de três testes destinados à validação das migrações e cinco testes de integração por meio de requisições *HTTP*. Todos os casos foram aprovados.

Os testes das migrações verificaram a execução da cadeia em uma base vazia, a criação das estruturas utilizadas pela aplicação e a aplicação das restrições previstas. Já os testes *HTTP* contemplaram fluxos envolvendo autenticação, autorização, controle de acesso e operações acadêmicas realizadas entre professor e estudante.

Quadro 18 - Resultado dos testes com PostgreSQL

| <b>Categoria</b>          | <b>Testes executados</b> | <b>Resultado</b> |
|---------------------------|--------------------------|------------------|
| Integração com PostgreSQL | 59                       | Aprovados        |

|                             |    |           |
|-----------------------------|----|-----------|
| Migrações do banco de dados | 3  | Aprovados |
| Integração HTTP             | 5  | Aprovados |
| Total                       | 67 | Aprovados |

Fonte: autor (2026).

A execução desses testes complementou os 351 casos do *backend* que não dependem do ambiente PostgreSQL, resultando em 418 testes distintos de *backend* executados e aprovados. Enquanto os testes convencionais verificaram principalmente regras de negócio e componentes em maior isolamento, os testes de integração permitiram avaliar a persistência real dos dados, as restrições do banco e fluxos que percorrem diferentes camadas da aplicação.

#### 8.4 AVALIAÇÃO HEURÍSTICA DE USABILIDADE

A avaliação da interface do RepolF foi realizada com base nas dez heurísticas de usabilidade propostas por Nielsen (1994). A inspeção teve como objetivo verificar a aderência das principais interfaces da plataforma aos princípios relacionados ao *feedback* do sistema, linguagem, controle do usuário, consistência, prevenção e tratamento de erros, organização das informações e eficiência da interação.

Foram consideradas interfaces representativas das funcionalidades destinadas aos diferentes perfis da plataforma. Para cada heurística, foram observados elementos e comportamentos da interface capazes de evidenciar sua aplicação no RepolF. As subseções seguintes apresentam os resultados da inspeção, acompanhados de evidências visuais extraídas da própria aplicação.

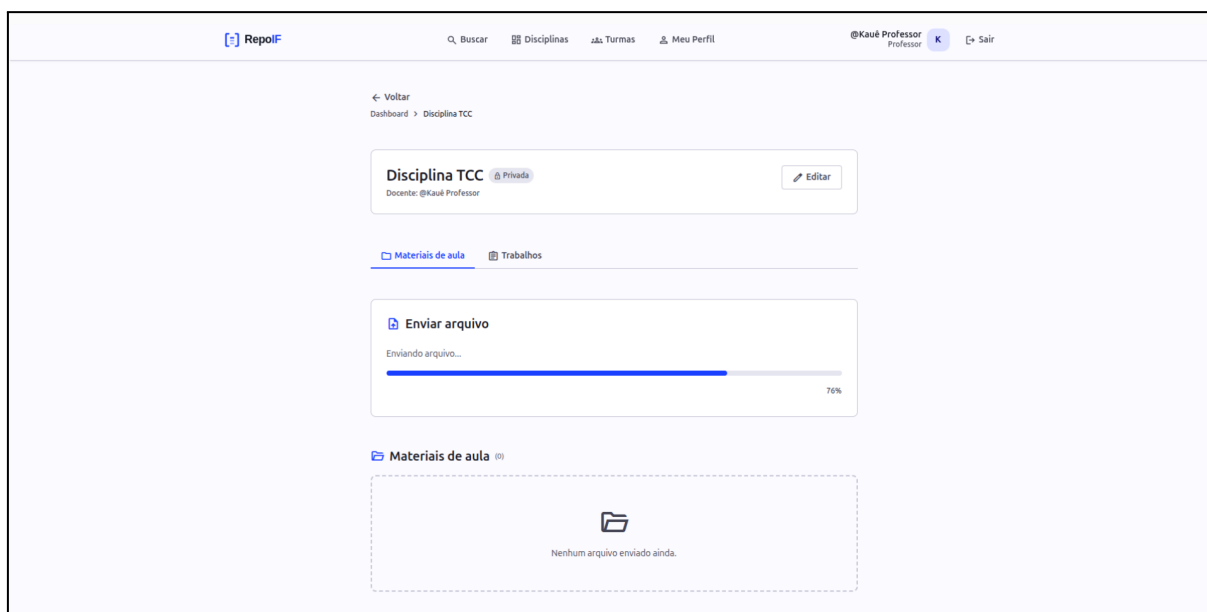
##### 8.4.1 Visibilidade do estado do sistema

A primeira heurística estabelece que o sistema deve manter o usuário informado sobre o que está acontecendo, fornecendo *feedback* adequado após suas ações (Nielsen, 1994).

No RepolF, esse princípio pode ser observado por meio das mensagens

apresentadas após operações realizadas pelo usuário, dos estados de carregamento e do bloqueio temporário de determinadas ações durante o processamento. Após operações como criação, edição, envio ou exclusão de registros, a interface apresenta retornos que permitem ao usuário compreender o resultado da ação executada.

Figura 17 - *Feedback* apresentado ao usuário após uma operação no RepolF



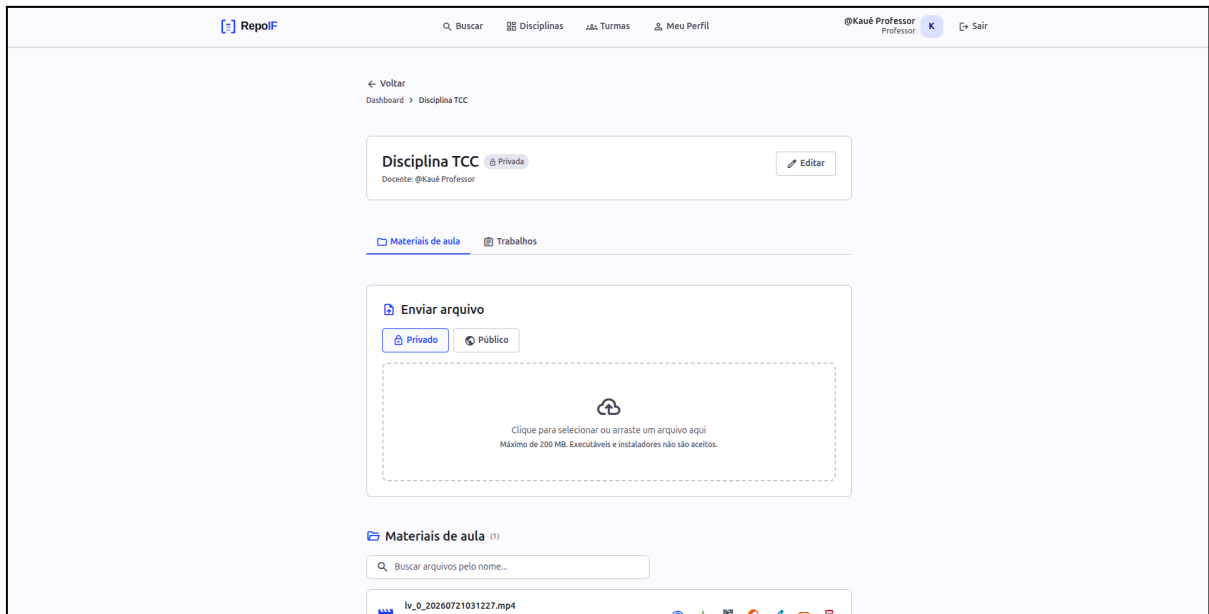
Fonte: autor (2026).

#### 8.4.2 Correspondência entre o sistema e o mundo real

Essa heurística recomenda que a interface utilize conceitos, termos e representações familiares ao usuário, evitando que o funcionamento do sistema dependa da compreensão de terminologias técnicas (Nielsen, 1994).

No RepolF, as interfaces utilizam predominantemente termos relacionados ao contexto acadêmico, como disciplina, turma, professor, estudante, atividade, material, entrega e prazo. Dessa forma, conceitos internos da implementação permanecem abstraídos, enquanto as funcionalidades são apresentadas de acordo com a terminologia esperada no ambiente educacional.

Figura 18 - Utilização de terminologia acadêmica nas interfaces do RepolF



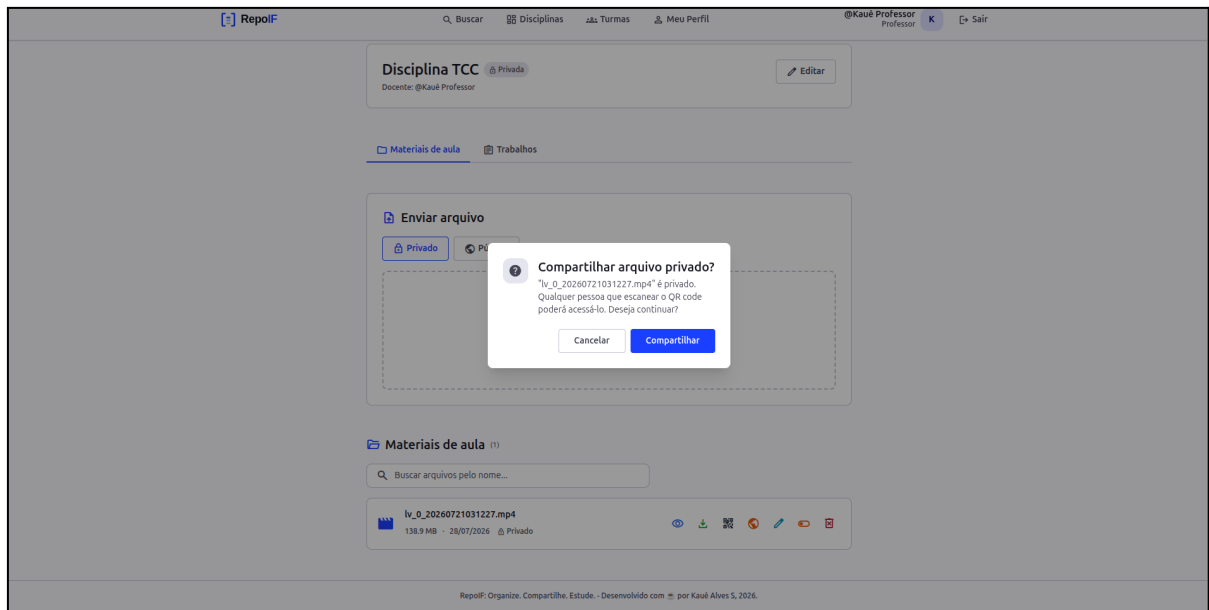
Fonte: autor (2026).

### 8.4.3 Controle e liberdade do usuário

A terceira heurística está relacionada à possibilidade de o usuário controlar suas ações e abandonar operações indesejadas antes que sejam concluídas (Nielsen, 1994).

No RepolF, esse princípio é aplicado principalmente em formulários, modais e operações que exigem confirmação. O usuário pode cancelar determinadas ações antes de sua execução definitiva, fechar interfaces modais e navegar entre as diferentes áreas da aplicação. Operações potencialmente destrutivas também utilizam mecanismos de confirmação, reduzindo a possibilidade de execução acidental.

Figura 19 - Controle do usuário em operação que exige confirmação



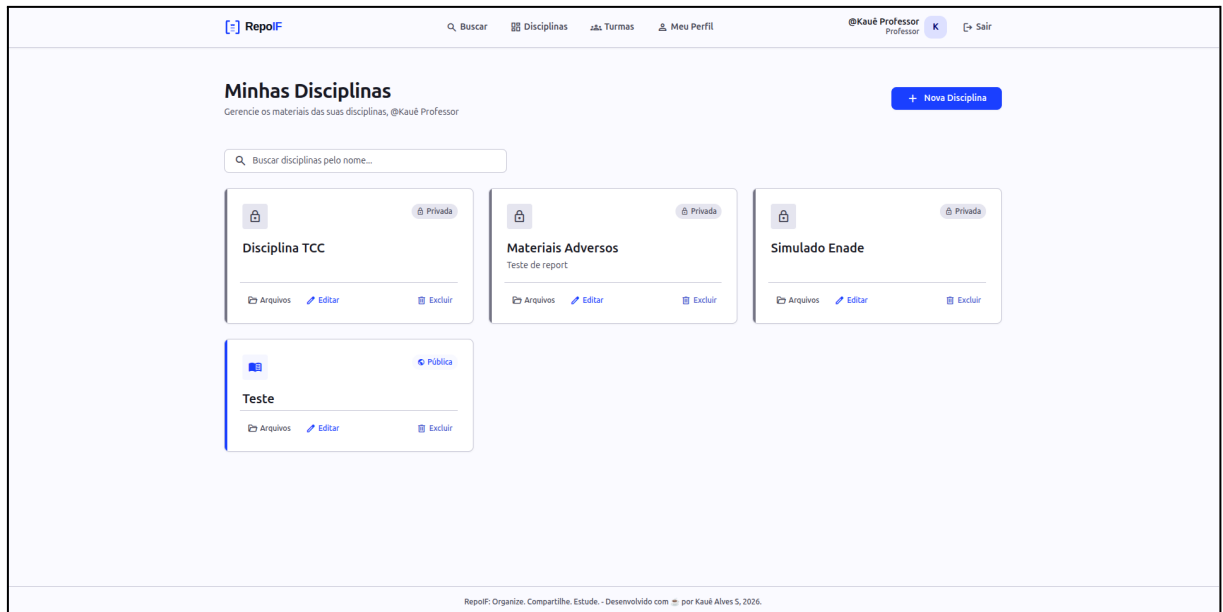
Fonte: autor (2026).

#### 8.4.4 Consistência e padrões

A consistência busca evitar que o usuário precise interpretar diferentes significados para elementos que executam funções semelhantes. Ações, componentes e terminologias devem manter padrões ao longo da aplicação (Nielsen, 1994).

No RepoIF, essa característica pode ser observada na reutilização de padrões visuais e comportamentais em botões, formulários, modais, mensagens de retorno e elementos de navegação. Ações semelhantes são apresentadas de forma consistente em diferentes módulos, contribuindo para que o usuário reconheça seu funcionamento ao navegar pela plataforma.

Figura 20 - Padronização dos elementos de interface do RepolF



Fonte: autor (2026).

#### 8.4.5 Prevenção de erros

A prevenção de erros busca evitar que situações problemáticas ocorram, em vez de depender somente da apresentação de uma mensagem após a execução de uma operação inválida (Nielsen, 1994).

No RepolF, são utilizados mecanismos como validação de formulários, campos obrigatórios, confirmação de ações críticas e bloqueio de determinadas operações durante seu processamento. A aplicação também restringe ações de acordo com o papel e as permissões do usuário, além de aplicar validações específicas em operações como submissões, convites, denúncias e envio de arquivos.

Figura 21 - Mecanismo de prevenção de erros em uma operação do RepolF

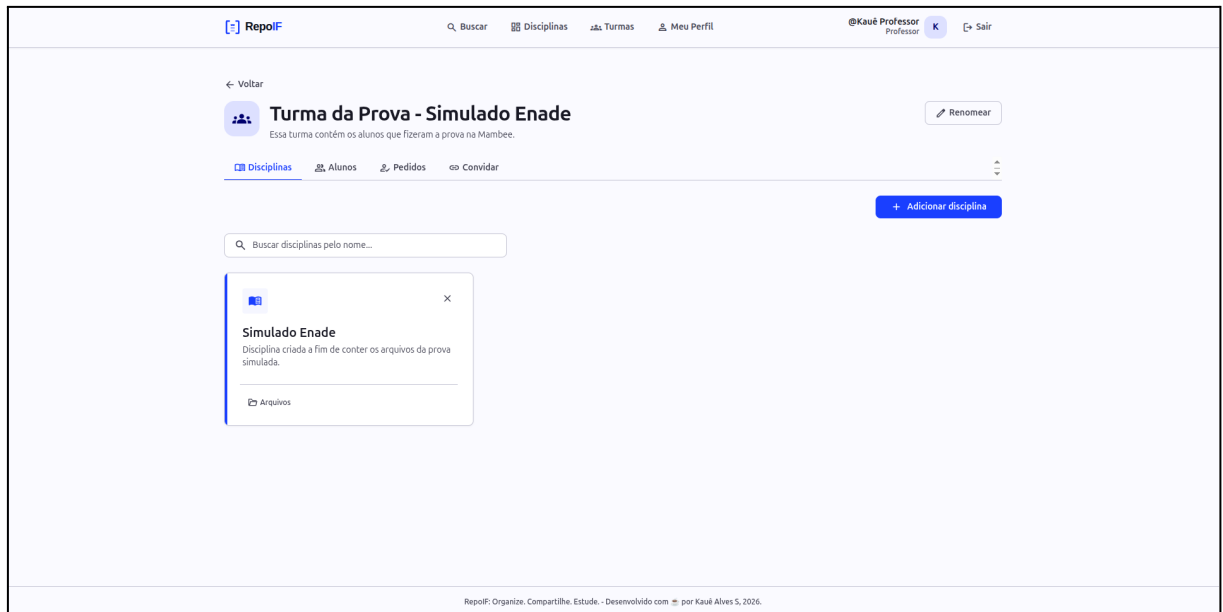
Fonte: autor (2026).

#### 8.4.6 Reconhecimento em vez de memorização

Essa heurística recomenda reduzir a quantidade de informações que o usuário precisa manter na memória durante a utilização do sistema. Elementos, opções e informações relevantes devem permanecer disponíveis sempre que forem necessários (Nielsen, 1994).

No RepolF, as informações relacionadas ao contexto das operações são apresentadas diretamente nas interfaces. Disciplinas, turmas, atividades, materiais e usuários são identificados visualmente, enquanto ações relevantes permanecem disponíveis junto aos respectivos elementos. Esse comportamento reduz a necessidade de o usuário memorizar informações provenientes de telas anteriores.

Figura 22 - Apresentação contextual de informações e ações no RepolF



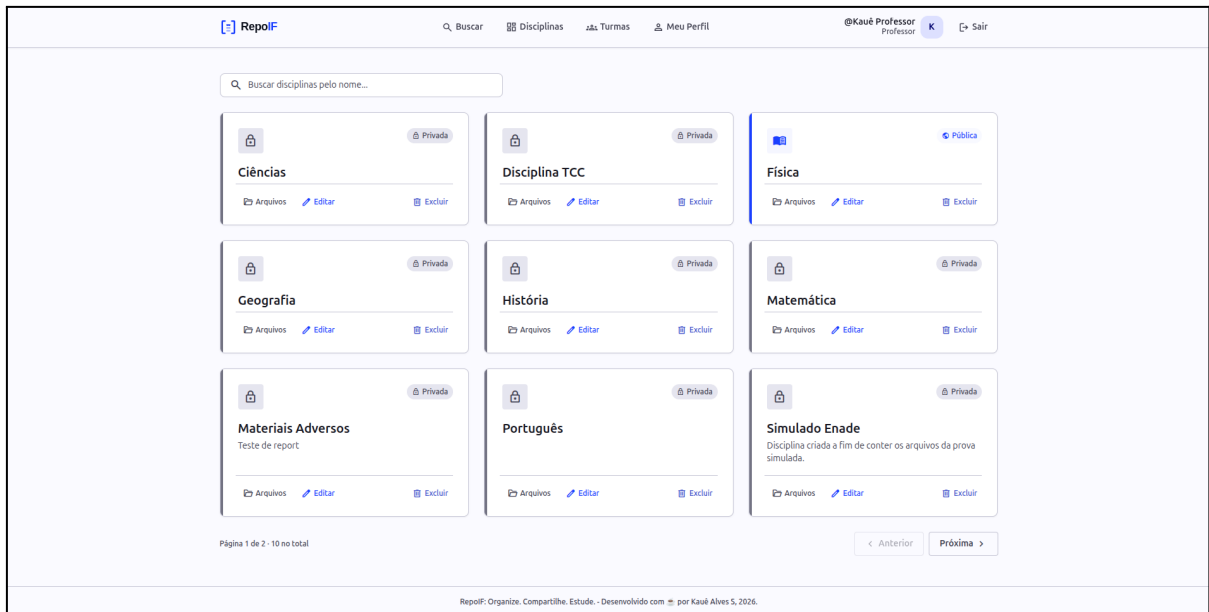
Fonte: autor (2026).

#### 8.4.7 Flexibilidade e eficiência de uso

A sétima heurística está relacionada à possibilidade de realizar tarefas de maneira eficiente, incluindo recursos que facilitem a interação à medida que a quantidade de informações ou a frequência de utilização aumenta (Nielsen, 1994).

No RepolF, recursos como pesquisa, filtros, paginação e acesso direto às ações associadas aos registros contribuem para facilitar a localização e o gerenciamento das informações. Esses mecanismos estão presentes em diferentes áreas da plataforma e reduzem a quantidade de etapas necessárias para localizar disciplinas, usuários, materiais e outros registros.

Figura 23 - Recursos de pesquisa e filtragem disponíveis no RepolF



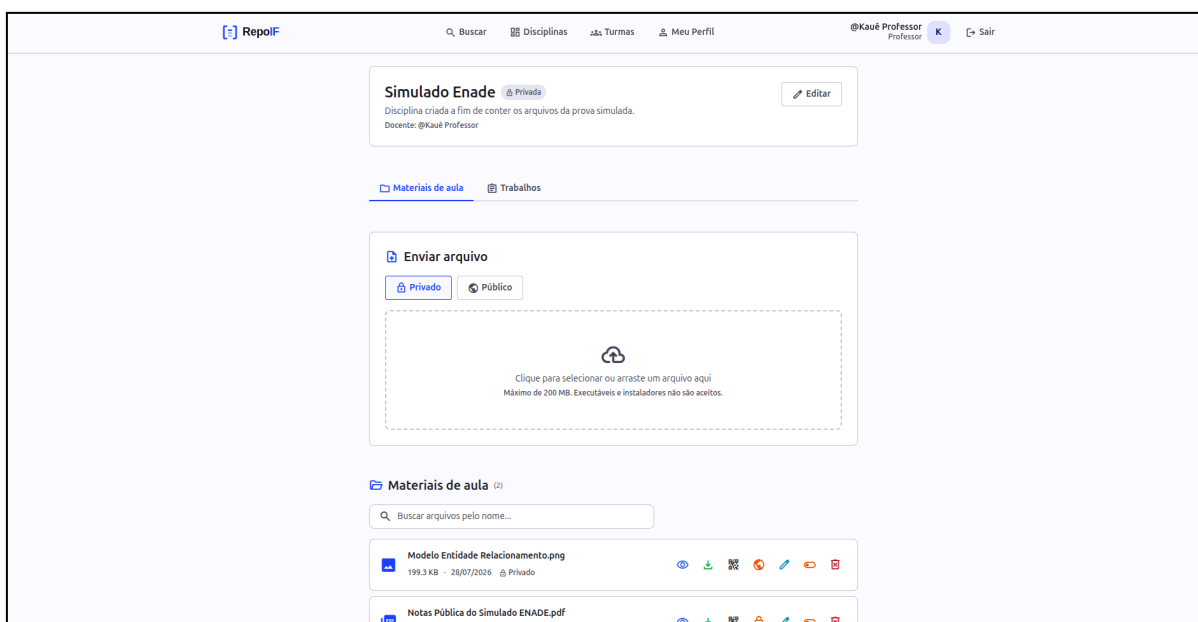
Fonte: autor (2026).

#### 8.4.8 Design estético e minimalista

A heurística de *design* estético e minimalista recomenda que a interface apresente somente informações relevantes para a realização das tarefas, evitando elementos que disputem desnecessariamente a atenção do usuário (Nielsen, 1994).

As interfaces do RepolF foram organizadas buscando manter hierarquia entre títulos, informações, conteúdos e ações disponíveis. Os recursos são distribuídos conforme o contexto de cada página, priorizando as informações relacionadas à tarefa realizada pelo usuário e evitando a exposição de elementos técnicos da implementação.

Figura 24 - Organização visual de uma das principais interfaces do RepolF



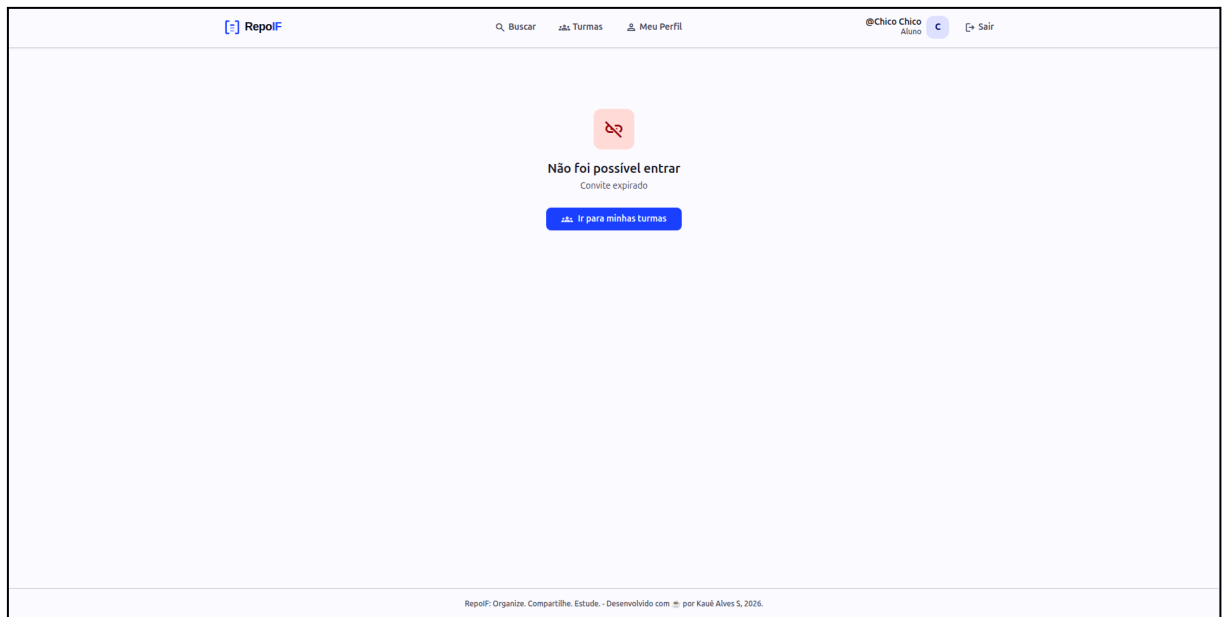
Fonte: autor (2026).

#### 8.4.9 Reconhecimento, diagnóstico e recuperação de erros

Essa heurística estabelece que as mensagens apresentadas diante de situações inválidas devem utilizar linguagem compreensível e auxiliar o usuário a identificar o problema e prosseguir com a utilização do sistema (Nielsen, 1994).

O RepolF apresenta mensagens específicas em diferentes situações de erro, evitando expor diretamente informações técnicas provenientes da implementação. Entre os casos tratados estão falhas de autenticação, formulários inválidos, operações não autorizadas e convites inválidos, permitindo ao usuário compreender a situação ocorrida e, quando aplicável, tomar uma ação para corrigi-la.

Figura 25 - Tratamento de situação de erro apresentado pelo RepolF



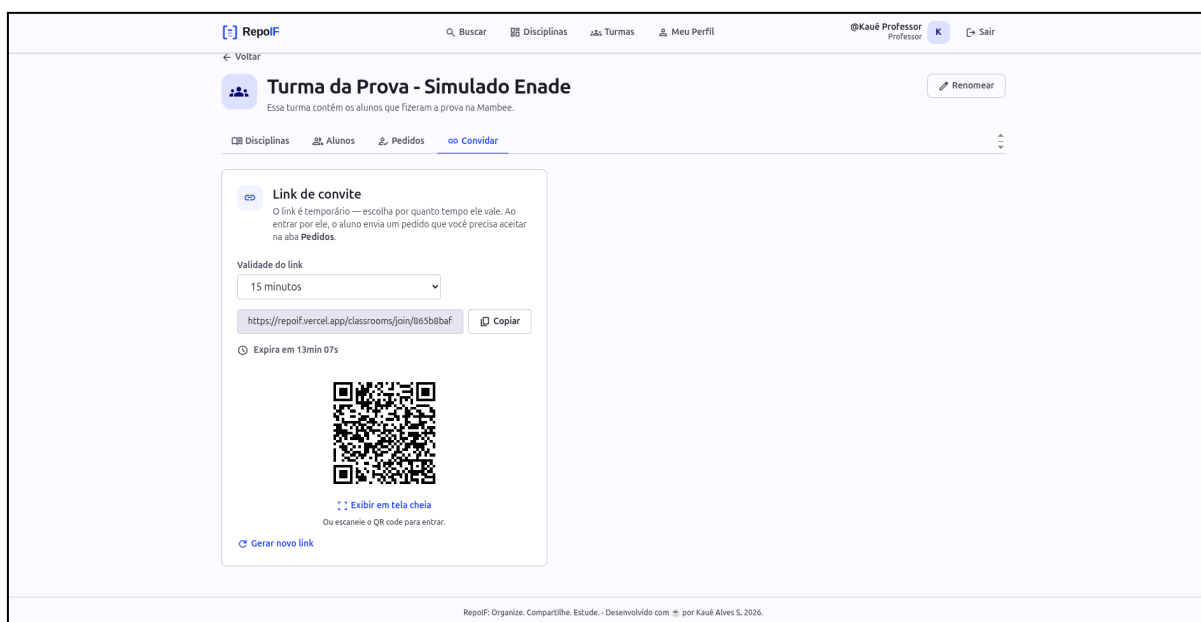
Fonte: autor (2026).

#### 8.4.10 Ajuda e documentação

A última heurística considera a necessidade de fornecer orientação ao usuário quando determinada funcionalidade não puder ser compreendida de maneira suficientemente clara apenas pelos elementos da interface (Nielsen, 1994).

No RepolF, a orientação é apresentada de forma contextual, por meio de rótulos, descrições, mensagens e informações associadas às operações em que explicações adicionais são necessárias. Essa abordagem busca permitir que as principais tarefas sejam compreendidas diretamente nas próprias interfaces, sem exigir consulta constante a documentação externa.

Figura 26 - Orientação contextual apresentada em uma funcionalidade do RepoIF



Fonte: autor (2026).

#### 8.4.11 Síntese da avaliação heurística

A inspeção das interfaces selecionadas evidenciou a aplicação dos dez princípios de usabilidade considerados na avaliação. Foram observados mecanismos destinados a informar o estado das operações, utilizar terminologia compatível com o contexto acadêmico, oferecer controle ao usuário, manter consistência entre as interfaces, prevenir e tratar erros e reduzir a necessidade de memorização. Também foram identificados recursos voltados à eficiência das tarefas, à organização visual das informações e à orientação contextual durante a utilização da plataforma.

Com base nas evidências registradas para cada uma das dez heurísticas, foram identificados elementos e comportamentos das interfaces compatíveis com os princípios de usabilidade considerados. Os resultados indicam, portanto, aderência das interfaces analisadas do RepoIF às heurísticas adotadas como referência neste trabalho.

Ressalta-se, contudo, que a inspeção foi realizada pelo próprio autor e desenvolvedor da aplicação. Essa condição pode influenciar a interpretação das interfaces em razão da familiaridade prévia com o funcionamento do sistema. Além

disso, a aderência às heurísticas não deve ser interpretada como comprovação absoluta da usabilidade da plataforma, uma vez que a avaliação não envolveu testes com usuários representativos.

## 8.5 ATENDIMENTO AOS OBJETIVOS

A análise conjunta da implementação, da validação técnica e da avaliação heurística permitiu verificar o atendimento dos principais objetivos estabelecidos para o desenvolvimento do RepolF.

A plataforma resultante disponibiliza recursos para gerenciamento e compartilhamento de materiais didáticos, organização de disciplinas e turmas, criação e acompanhamento de atividades, controle de acesso entre diferentes perfis e administração e moderação da aplicação.

A arquitetura implementada também permitiu separar os dados estruturados dos arquivos binários, utilizando PostgreSQL e *Cloudflare R2* para finalidades distintas. Os mecanismos de autenticação, autorização e validação foram aplicados às operações protegidas, enquanto os testes automatizados permitiram verificar diferentes regras de negócio, comportamentos da interface e situações de erro.

O Quadro 19 relaciona os principais objetivos específicos definidos para o desenvolvimento da solução às evidências produzidas ao longo do trabalho.

Quadro 19 - Relação entre os objetivos específicos e os resultados obtidos

| Objetivo                                                                                                 | Evidência de atendimento                                                                 |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Analisar conceitos relacionados às tecnologias digitais, materiais didáticos e organização da informação | Fundamentação teórica desenvolvida a partir da literatura selecionada                    |
| Levantar e documentar os requisitos da aplicação                                                         | Requisitos funcionais, não funcionais e regras de negócio especificados                  |
| Definir a arquitetura e a modelagem do sistema                                                           | Arquitetura da aplicação, casos de uso e modelagem dos dados                             |
| Implementar autenticação e controle de acesso                                                            | Cadastro, verificação de conta, autenticação, recuperação de senha e controle por papéis |

|                                                                                  |                                                                                                                                                                                        |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Implementar o gerenciamento de disciplinas e materiais                           | Criação e gerenciamento de disciplinas, armazenamento, consulta e controle de materiais                                                                                                |
| Implementar o gerenciamento de turmas                                            | Criação de turmas, associação de disciplinas, gerenciamento de integrantes e convites                                                                                                  |
| Implementar atividades e entregas                                                | Criação de atividades, envio de arquivos, acompanhamento das entregas e controle de reenvio                                                                                            |
| Implementar recursos administrativos e de moderação                              | Gerenciamento de usuários e materiais, denúncias e funcionalidades administrativas                                                                                                     |
| Aplicar mecanismos de segurança e consistência                                   | Validações, controle de acesso, proteção das operações com arquivos e mecanismos de consistência entre banco e armazenamento                                                           |
| Realizar a validação técnica da aplicação                                        | 418 testes de <i>backend</i> e 159 testes de <i>frontend</i> aprovados, totalizando 577 casos distintos, incluindo testes de integração com PostgreSQL, migrações e fluxos <i>HTTP</i> |
| Avaliar a usabilidade das interfaces                                             | Inspeção das principais interfaces e fluxos com base nas dez heurísticas de Nielsen, com análise textual e evidências visuais de aderência apresentadas na Seção 8.4.                  |
| Documentar o desenvolvimento e analisar o atendimento aos requisitos e objetivos | Capítulos 5 a 8, documentação técnica, matriz entre objetivos, implementação e resultados                                                                                              |

Fonte: autor (2026).

Os resultados apresentados neste capítulo indicam que os objetivos estabelecidos para o desenvolvimento e a avaliação do RepolF foram contemplados pelas etapas de implementação e validação descritas. Entretanto, tais resultados restringem-se à verificação do funcionamento técnico da solução e ao atendimento do escopo proposto, não constituindo evidências de impacto pedagógico ou de satisfação dos usuários, aspectos que demandariam estudos específicos.

## 9 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento e a validação técnica do RepolF, uma plataforma web voltada ao gerenciamento e ao compartilhamento centralizado de materiais didáticos no contexto do Instituto Federal de Educação, Ciência e Tecnologia do Piauí - Campus Picos. A proposta partiu da problemática relacionada à dispersão de conteúdos acadêmicos entre diferentes meios digitais e buscou oferecer uma estrutura na qual materiais, disciplinas, turmas e atividades pudessem ser organizados e gerenciados de forma integrada.

A questão de pesquisa buscou compreender como uma plataforma web poderia contribuir para a centralização, a organização e a recuperação de materiais didáticos. A solução desenvolvida oferece uma resposta técnica à questão proposta ao reunir os conteúdos em um ambiente estruturado, associando-os ao contexto acadêmico em que são utilizados e disponibilizando mecanismos de pesquisa, consulta e controle de acesso. O RepolF também diferencia as operações permitidas aos perfis de estudante, professor e administrador, permitindo que os recursos disponibilizados sejam acessados e gerenciados de acordo com as responsabilidades definidas para cada usuário.

O desenvolvimento contemplou funcionalidades relacionadas à autenticação e ao gerenciamento de contas, criação e organização de disciplinas, publicação e gerenciamento de materiais, organização de turmas, ingresso de estudantes, criação de atividades, envio e controle de entregas, além de recursos administrativos e de moderação. Embora a plataforma incorpore funcionalidades relacionadas a turmas e atividades, seu escopo permanece concentrado no gerenciamento e no compartilhamento estruturado de materiais didáticos, não se propondo a substituir integralmente sistemas acadêmicos ou Ambientes Virtuais de Aprendizagem.

Do ponto de vista arquitetural, a solução foi estruturada segundo o modelo cliente-servidor, com separação entre frontend, backend, banco de dados e armazenamento de objetos. O frontend concentra as interfaces e a interação com os usuários, enquanto o backend é responsável pela aplicação das regras de negócio, autenticação, autorização e integração com os demais componentes. Os dados estruturados e os metadados são persistidos no PostgreSQL, enquanto os arquivos

binários são armazenados no Cloudflare R2. A arquitetura também incorpora mecanismos específicos para autorização e confirmação de operações com arquivos, bem como uma estratégia persistente para tratamento de objetos cuja remoção do armazenamento não possa ser concluída imediatamente.

A validação técnica constituiu uma etapa relevante para verificar o comportamento da implementação além da simples demonstração das funcionalidades. No *backend*, foram executados e aprovados 418 casos de teste distintos, incluindo testes das regras de negócio, componentes da aplicação, integração com PostgreSQL, migrações e fluxos *HTTP*. No *frontend*, foram executados e aprovados 159 testes voltados principalmente ao comportamento das interfaces e às interações disponibilizadas aos usuários. Dessa forma, foram executados 577 casos de teste automatizados distintos, todos aprovados na versão avaliada. Esses resultados fornecem evidências de que os comportamentos contemplados pelas suítes apresentaram os resultados esperados nos cenários verificados, incluindo operações válidas, entradas inválidas, restrições de acesso, regras de negócio e situações relacionadas à persistência e à integridade dos dados.

A análise das interfaces complementou a validação técnica por meio da aplicação das dez heurísticas de usabilidade de Nielsen. Para cada princípio, foram identificados elementos e comportamentos da aplicação capazes de evidenciar sua utilização, abrangendo aspectos relacionados à visibilidade do estado do sistema, correspondência com o contexto dos usuários, controle e liberdade, consistência, prevenção de erros, reconhecimento em vez de memorização, flexibilidade e eficiência de uso, organização visual, tratamento de erros e orientação contextual. As evidências registradas indicaram aderência das interfaces analisadas aos princípios adotados como referência na avaliação.

Os resultados obtidos permitem considerar atendido o objetivo geral de desenvolver e validar tecnicamente o RepolF, assim como os objetivos específicos relacionados ao levantamento e à especificação dos requisitos, definição da arquitetura e da modelagem dos dados, implementação das funcionalidades, aplicação dos mecanismos de segurança, realização dos testes automatizados e avaliação das interfaces. A documentação das etapas de requisitos, modelagem,

arquitetura, implementação e validação também possibilitou estabelecer uma relação entre as características previstas para a solução e aquelas efetivamente presentes na versão desenvolvida.

Diante das evidências obtidas, conclui-se que o RepolF constitui uma solução computacional funcional para o escopo estabelecido neste trabalho, oferecendo mecanismos para centralização, organização, gerenciamento, compartilhamento e recuperação de materiais didáticos, associados a recursos de turmas, atividades, controle de acesso, administração e moderação. Sua implementação permitiu aplicar de forma integrada conceitos de Engenharia de Software relacionados à especificação de requisitos, modelagem de dados, arquitetura de aplicações web, segurança, persistência, armazenamento de arquivos, testes automatizados e avaliação de interfaces. Dessa forma, o trabalho alcançou os objetivos definidos e resultou em uma plataforma tecnicamente implementada, documentada e submetida a procedimentos sistemáticos de verificação e avaliação.

## 9.1 TRABALHOS FUTUROS

As limitações identificadas durante a realização deste trabalho estabelecem possibilidades para a continuidade da avaliação e da evolução do RepolF. A principal delas está relacionada à ausência de avaliações envolvendo usuários representativos da comunidade acadêmica. A validação realizada concentrou-se nos aspectos técnicos da aplicação e na inspeção heurística de suas interfaces, sem a participação de professores e estudantes em testes estruturados de utilização. Dessa forma, uma continuidade relevante consiste na realização de estudos com usuários, permitindo investigar aspectos como facilidade de uso, satisfação, aceitação da plataforma e dificuldades encontradas durante a execução de tarefas.

A inspeção heurística também apresenta como limitação o fato de ter sido conduzida pelo próprio autor e desenvolvedor do RepolF. Embora a análise tenha utilizado critérios previamente estabelecidos e evidências extraídas das interfaces, a familiaridade com o funcionamento da aplicação pode influenciar a interpretação dos elementos avaliados. Como continuidade, a inspeção poderá ser realizada por outros avaliadores, possibilitando a comparação entre diferentes análises e ampliando a independência do processo de avaliação da usabilidade.

Outra possibilidade está relacionada à avaliação da recuperação dos materiais. O presente trabalho demonstrou a existência de mecanismos de organização, pesquisa, filtragem e contextualização dos conteúdos, mas não realizou experimentos comparativos para mensurar o tempo, o número de interações ou a taxa de sucesso na localização de materiais. Estudos posteriores poderão avaliar esses aspectos com usuários representativos e comparar a utilização do RepolF com outras formas de compartilhamento empregadas no contexto acadêmico.

Os testes automatizados realizados também estão limitados aos comportamentos e condições representados pelos casos implementados. Embora tenham sido executados 577 testes distintos e tenham sido incluídos cenários de integração com PostgreSQL, migrações e requisições HTTP, a validação não contempla todas as condições que podem ocorrer em um ambiente de produção. Dessa forma, podem ser conduzidas avaliações adicionais relacionadas ao desempenho, à escalabilidade e ao comportamento da aplicação sob diferentes volumes de usuários, requisições e arquivos armazenados.

Por fim, a evolução do RepolF poderá considerar novas funcionalidades e integrações desde que permaneçam coerentes com a finalidade da plataforma. Possíveis expansões podem envolver integração com serviços institucionais, aprimoramento dos mecanismos de organização e recuperação dos conteúdos e aperfeiçoamentos derivados de avaliações realizadas com usuários. Essas possibilidades devem ser consideradas como continuidade do trabalho desenvolvido, sem alterar o fato de que os resultados apresentados nesta pesquisa estão restritos à versão e aos procedimentos de avaliação efetivamente documentados.

## REFERÊNCIAS

- ALVES, M. A. B.; MEDINA, T. S.; SARTORI, V.; DAL FORNO, L. F. *Compartilhamento de informações e conhecimento: desafios educacionais, tecnológicos e familiares em tempos de pandemia*. 2022. Disponível em: [https://d1wqtxts1xzle7.cloudfront.net/113368503/farias\\_2C\\_8\\_RCD\\_34\\_-\\_p\\_136-151.pdf](https://d1wqtxts1xzle7.cloudfront.net/113368503/farias_2C_8_RCD_34_-_p_136-151.pdf). Acesso em: 18 fev. 2026.
- CLOUDFLARE. *Presigned URLs: Cloudflare R2 documentation*. 2026. Disponível em: <https://developers.cloudflare.com/r2/api/s3/presigned-urls>. Acesso em: 30 jul. 2026.
- COLUSSO, P. R.; ASTUDILLO, M. V. *A escolha e o uso de material didático digital por professores da educação superior: a evolução das TDIC no ambiente virtual de aprendizagem*. Research, Society and Development, v. 12, n. 3, 2023. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/40395>. Acesso em: 18 fev. 2026.
- GOMES, F. C. S. *As tecnologias digitais como ferramenta para a produção e melhoria do material didático no contexto escolar*. Revista Ibero-Americana de Humanidades, Ciências e Educação, [S. l.], v. 11, n. 8, p. 1010–1020, 2025. DOI: 10.51891/rease.v11i8.20613. Disponível em: <https://periodicorease.pro.br/rease/article/view/20613>. Acesso em: 18 fev. 2026.
- JEST. *Getting started*. [S. l.], [s. d.]. Disponível em: <https://jestjs.io/docs/getting-started>. Acesso em: 30 jul. 2026.
- LAUDON, K. C.; LAUDON, J. P. *Sistemas de informação gerenciais: Administrando a empresa digital*. 17. ed. 2023. Disponível em: [https://books.google.com.br/books?hl=pt-BR&lr=&id=NlehEAAQBAJ&oi=fnd&pg=PT14&dq=laudon+e+laudon&ots=f3hsU2uXt3&sig=gljoQKeeXrIKT6Z\\_4UNd4vEzKEk&redir\\_esc=y#v=onepage&q=laudon%20e%20laudon&f=false](https://books.google.com.br/books?hl=pt-BR&lr=&id=NlehEAAQBAJ&oi=fnd&pg=PT14&dq=laudon+e+laudon&ots=f3hsU2uXt3&sig=gljoQKeeXrIKT6Z_4UNd4vEzKEk&redir_esc=y#v=onepage&q=laudon%20e%20laudon&f=false). Acesso em: 18 fev. 2026.
- NESTJS. *First steps*. [S. l.], [s. d.]. Disponível em: <https://docs.nestjs.com/first-steps>. Acesso em: 30 jul. 2026.
- NIELSEN, J. *Usability engineering*. Boston: Academic Press, 1994.
- POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL 18.4 documentation*. 2026. Disponível em: <https://www.postgresql.org/docs/current>. Acesso em: 30 jul. 2026.
- REACT. *React reference overview*. [S. l.], [s. d.]. Disponível em: <https://react.dev/reference/react>. Acesso em: 30 jul. 2026.
- RODRIGUES, R. S.; TAGA, V.; VIEIRA, E. M. F. *Repositórios educacionais: estudos preliminares para a Universidade Aberta do Brasil*. Perspectivas em Ciência da Informação, v. 16, n. 3, 2011. Disponível em: <https://www.scielo.br/j/pci/a/bkscLDR4hV4CFDrZxHKyPzF/?lang=pt>. Acesso em: 18 fev. 2026.

SANTOS, G. E. dos; OLIVEIRA, G. V. de; OLIVEIRA, K. A. V. de; SOUSA, T. O. de. *Gestão eletrônica de materiais didáticos no ambiente escolar*. 2022. Disponível em: <http://ric-cps.eastus2.cloudapp.azure.com/handle/123456789/11108>. Acesso em: 18 fev. 2026.

SANTOS, S. R. dos. *Plataformas digitais: o uso das plataformas digitais na educação*. 2023. Disponível em: <https://repositoriocopia.utfpr.edu.br/jspui/handle/1/35900>. Acesso em: 18 fev. 2026.

SOMMERVILLE, Ian. Engenharia de software. Tradução de Ivan Bosnic e Kalinka G. de O. Gonçalves. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Disponível em: <https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>. Acesso em: 27 fev. 2026.

TESTING LIBRARY. *React Testing Library*. [S. l.], [s. d.]. Disponível em: <https://testing-library.com/docs/react-testing-library/intro>. Acesso em: 30 jul. 2026.

TYPEORM. *Getting started*. [S. l.], [s. d.]. Disponível em: <https://typeorm.io/docs/getting-started>. Acesso em: 30 jul. 2026.

VITE. *Getting started*. [S. l.], [s. d.]. Disponível em: <https://vite.dev/guide>. Acesso em: 30 jul. 2026.

VITEST. *Getting started*. [S. l.], [s. d.]. Disponível em: <https://vitest.dev/guide>. Acesso em: 30 jul. 2026.