

UTFPR-UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Bacharelado em Engenharia de Software - 6º Período

DISCIPLINA: *Oficina de Integração 1 - ES46F-ES61*

PROFESSOR: *Eduardo Cotrin Teixeira*

Documento de

Projeto de Software

GERAS

**Carlos Eduardo M. D. Lopes, Bruno Meleiro Dutra,
Pedro Lucas S. Mota, Thiago Gardelli**

Cornélio Procópio

2025

Sumário

1	INTRODUÇÃO	2
1.1	Contexto	2
1.2	Justificativa	2
1.3	Proposta	2
1.4	Organização do Documento	2
2	DESCRIÇÃO GERAL DO SISTEMA	3
2.1	Objetivos (Gerais e Específicos)	3
2.2	Limites e Restrições	3
2.3	Descrição dos Usuários do Sistema	3
3	DESENVOLVIMENTO DO PROJETO	4
3.1	Tecnologias e ferramentas	4
3.2	Metodologia de desenvolvimento	4
3.3	Cronograma previsto	4
4	REQUISITOS DO SISTEMA	5
4.1	Requisitos Funcionais	5
4.2	Requisitos Não-funcionais	5
4.3	Diagramas de Casos de Uso	5
4.4	Protótipos de Telas	6
5	ANÁLISE DO SISTEMA	7
6	IMPLEMENTAÇÃO	8
6.1	Descrição do código	8
6.2	Implantação	8
6.3	Telas principais	8
7	CONSIDERAÇÕES FINAIS	9
8	BIBLIOGRAFIA	10

1 Introdução

1.1 Contexto

O TEDI (Tecnologia e Educação Digital para terceira Idade), projeto de extensão da UTFPR é voltado ao ensino de tecnologia para a terceira idade. Nele, eventos como palestras e oficinas ensinando sobre smartphones e computadores são oferecidos gratuitamente, além de visitas ao CRAS (Centro de Referência de Assistência Social) com o mesmo propósito. Tendo isso em mente, o projeto produz materiais analógicos e digitais personalizados para idosos, que vão desde jogos até apostilas e textos educativos. Todavia, todos esses documentos ficam muito descentralizados, visto que as produções dos materiais ficam armazenadas pelos membros que as produziram, dificultando a vida tanto dos membros do TEDI, quando têm que procurar algum arquivo, quanto dos idosos e demais públicos, que não conseguem achar uma fonte de acesso aos materiais do TEDI.

Esses materiais são conhecidos dentro do meio como REAs, Recursos Educacionais Abertos, que de acordo com a Unesco são: “materiais de aprendizagem, ensino e pesquisa em qualquer formato e meio que residam em domínio público ou estejam protegidos por direitos autorais que foram lançados sob uma licença aberta, que permitem acesso gratuito, reutilização, reaproveitamento, adaptação e redistribuição por terceiros”¹.

Pensando nisso, será feito um repositório educacional de materiais para idosos, cujo papel será armazenar materiais específicos para a terceira idade e organizar para que possa ser aberto ao público com segurança e de maneira personalizada, trazendo mais acessibilidade para as necessidades especiais do público. A princípio o conteúdo seria inteiramente do TEDI, mas com possibilidade de expansão para mais projetos de outras instituições.

1.2 Justificativa

Devido à dispersão dos acervos e à ausência de um repositório centralizado, parte do material produzido pelo TEDI encontra-se inacessível ou com acesso restrito, o que prejudica tanto a atuação dos membros do projeto quanto o atendimento ao público idoso. Integrantes do TEDI, incluindo a Profa. Dra. Camila Dias de Oliveira Sestito (orientadora do projeto), relataram episódios em que participantes de oficinas ou palestras solicitaram materiais que não puderam ser disponibilizados na ocasião por estarem armazenados em locais distintos (arquivos pessoais, pastas em nuvem sem organização, mensagens de aplicativos, dispositivos locais) ou por problemas de permissão e indexação.

Essa fragmentação reduz o alcance e a reutilização dos recursos produzidos (REAs — Recursos Educacionais Abertos), aumenta o retrabalho e dificulta a manutenção e atualização dos conteúdos. Ademais, as necessidades específicas do público idoso (acessibilidade visual,

navegabilidade simplificada e formatos adaptáveis) exigem que os materiais estejam organizados e apresentados de modo acessível. Diante disso, justifica-se a implementação de um repositório web centralizado que garanta acesso controlado para os administradores do TEDI e acesso público e acessível para usuários externos, preservando a integridade dos arquivos e maximizando a disponibilidade dos REAs.

1.3 Proposta

Uma vez que o maior problema tem sido a dificuldade de organização, buscamos entender os requisitos e percebemos que a melhor solução seria implementar um repositório de REAs (Recursos Educacionais Acadêmicos) para o TEDI. Ele será uma plataforma web, acessada via navegador, que armazenará apostilas digitais, jogos educativos e qualquer tipo de material voltado à terceira idade. Em um primeiro momento o software será dividido entre o lado do administrador, que poderá fazer upload de conteúdo (a princípio somente o TEDI terá esse acesso, com possibilidade de expansão para outros projetos de outras instituições futuramente) e o lado do usuário, que poderá buscar acessar e baixar os conteúdos oferecidos.

1.4 Organização do Documento

O presente documento foi estruturado de forma a guiar o desenvolvimento do sistema GERAS de maneira clara e progressiva. O Capítulo 1 apresenta o contexto, a justificativa e a proposta do projeto, além de explicitar sua motivação. No Capítulo 2, descrevem-se os objetivos gerais e específicos, bem como as limitações do sistema e o perfil dos usuários que dele farão uso. Em seguida, o Capítulo 3 aborda o processo de desenvolvimento, detalhando as tecnologias escolhidas, a metodologia aplicada — baseada no Kanban — e o cronograma previsto para as entregas. O Capítulo 4 reúne os requisitos funcionais e não funcionais, complementados por diagramas de casos de uso e protótipos de interface. O Capítulo 5 trata da análise do sistema, com os artefatos de modelagem necessários para a compreensão da solução proposta. O Capítulo 6 descreve a implementação, incluindo a organização do código, o processo de implantação e a apresentação das telas principais. Por fim, o Capítulo 7 reúne as considerações finais acerca dos resultados obtidos, e o Capítulo 8 apresenta a bibliografia consultada.

2 Descrição Geral do Sistema

2.1 Objetivos (Gerais e Específicos)

2.1.1 Foco do projeto

O projeto GERAS tem como foco a criação de um repositório web de Recursos Educacionais Abertos (REAs) voltado prioritariamente ao público da terceira idade atendido pelo projeto TEDI. A solução será uma plataforma acessível, simples e centralizada para organização, busca, visualização e download de materiais educativos (apostilas, jogos, textos, etc.), reduzindo a descentralização atualmente observada na produção de conteúdos do TEDI.

2.1.2 Delimitação da solução

A solução a ser desenvolvida contempla apenas uma aplicação web, acessível via navegador, inicialmente com dois perfis principais: administradores (apenas membros do TEDI, responsáveis pelo upload e gerenciamento de conteúdo) e usuários externos (idosos, voluntários, parceiros institucionais) que poderão pesquisar e baixar os REAs. Não estão previstos, nesta primeira etapa, recursos de edição colaborativa online de documentos, integração com múltiplos provedores de armazenamento externos ou versão mobile nativa; a hospedagem será realizada pela infraestrutura do TEDI e as funcionalidades prioritárias deverão ser entregues até 06/12.

2.1.3 Objetivo geral

Desenvolver e entregar uma plataforma web (GERAS) que centralize, organize e disponibilize Recursos Educacionais Abertos produzidos pelo TEDI, garantindo acessibilidade e facilidade de uso para o público idoso, além de conformidade com a legislação aplicável sobre REAs.

2.1.4 Objetivos específicos

1. Projetar e implementar a interface da aplicação com foco em acessibilidade e usabilidade para idosos (navegação clara, fontes legíveis, alto contraste e controles simplificados).
2. Implementar autenticação e autorização para administradores do TEDI, garantindo que apenas usuários autorizados possam inserir e gerenciar conteúdos.
3. Desenvolver o módulo de upload e armazenamento de arquivos, incluindo salvamento de metadados (título, autor, data, descrição, tipo de material) em banco de dados PostgreSQL.
4. Implementar mecanismo de busca eficiente que permita pesquisa por texto livre e por metadados, com paginação dos resultados.
5. Disponibilizar funcionalidades de visualização e download dos REAs para usuários externos, preservando a integridade dos arquivos.

6. Garantir conformidade com a (Lei nº 14.198/2021)² sobre Recursos Educacionais Abertos e com requisitos mínimos de acessibilidade web.
7. Executar testes de usabilidade com representantes do público-alvo (pelo menos 5 usuários idosos) e incorporar melhorias de interface apontadas nas sessões de teste.
8. Realizar testes automatizados básicos (unitários e de integração) das funcionalidades críticas (autenticação, upload, busca, download) e testes manuais de regressão.
9. Documentar o sistema (guia de implantação, manuais de uso para administradores e usuários, e documentação do código) e preparar a entrega para hospedagem no ambiente do TEDI.
10. Planejar e executar a implantação da aplicação no ambiente de hospedagem do TEDI, incluindo configuração do banco de dados e políticas básicas de backup.

2.2 Limites e Restrições

A aplicação deverá ser oferecida apenas via web, com acesso por meio de navegadores e estar de acordo com a (Lei nº 14.198)², que regulamenta os recursos educacionais abertos (REAs). A interface e toda a usabilidade serão projetadas pensando primariamente em acessibilidade, uma vez que o público alvo não necessariamente tem um grande contato com a tecnologia. Em relação ao desenvolvimento, todas as funcionalidades principais (marcadas como prioridade alta) deverão ser entregues até dia 06/12. As tecnologias usadas serão Node.JS, HTML, CSS, Javascript e a hospedagem ficará por conta do TEDI.

2.3 Descrição dos Usuários do Sistema

Esta seção descreve os perfis de usuários previstos para a versão inicial da plataforma.

- **Administrador (membro do TEDI)**

Papel: gerenciar o acervo e as configurações do sistema.

Atribuições principais: realizar upload de conteúdo, editar metadados, remover registros, gerenciar contas administrativas, revisar logs de atividade e configurar políticas básicas (ex.: periodicidade de backup). Permissões: acesso total ao painel administrativo e às funcionalidades de gestão de conteúdo.

- **Consulta pública / Usuário externo**

Papel: consultar e utilizar os REAs disponibilizados.

Atribuições principais: pesquisar por título/metadados, visualizar prévias, efetuar download dos materiais, utilizar filtros de acessibilidade. Permissões: acesso às funcionalidades de consulta e download; não tem permissão para editar ou remover conteúdos.

- **Usuário colaborador**

Papel: usuário com conta criada e privilégios intermediários (por exemplo, professores parceiros).

Atribuições principais: enviar propostas de conteúdo para revisão, gerenciar conteúdos submetidos por si, participar de workflows de aprovação (caso implementado). Permissões: depende do nível concedido por administradores, mais geralmente podendo apenas cadastrar um material, editando e excluindo apenas com permissão do administrador.

Observação: na primeira fase a criação de contas será restrita a administradores — usuários externos não precisarão se registrar para consultar conteúdos.

3 Desenvolvimento do Projeto

3.1 Tecnologias e ferramentas

3.1.1 Linguagem de Programação

- O sistema utiliza HTML, CSS e Javascript para a criação e manutenção da interface do usuário.
- O sistema também utiliza de Node.js para fazer a conexão da interface do usuário com o servidor e o sistema de gerenciamento dos dados dos usuários.

3.1.2 Banco de dados

- Para gerenciamento do banco de dados, o sistema utiliza do PostgreSQL pela facilidade de implementação e manutenção no sistema.

3.1.3 Modelagem

- A modelagem foi feita utilizando as ferramentas draw.io e dbdiagram.io

3.1.4 Gerenciamento e organização

- A organização interna foi feita utilizando Github para o gerenciamento do código
- Whatsapp para comunicação entre os membros.
- Google Meet para reuniões com stakeholders e reuniões internas do time.

3.2 Metodologia de desenvolvimento

Para o desenvolvimento do GERAS adota-se uma abordagem ágil leve (Scrumban simplificado) adaptada a uma equipe de quatro integrantes. A estratégia prioriza entregas pequenas e frequentes, visibilidade do trabalho e responsabilidade clara por componente, sem sobrecarregar o time com cerimônias extensas. A metodologia define papéis, um fluxo de trabalho enxuto, critérios mínimos de qualidade e métricas básicas.

Papéis e responsabilidades

- **Product Owner (PO) — Carlos Eduardo**
Responsável pela priorização do backlog, relacionamento com stakeholders (TEDI, CRAS), validação funcional e aceite das entregas. Mantém a visão do produto e define prioridades semanais.
- **Tech Lead / Backend — Thiago Gardelli**
Coordena decisões de arquitetura, integrações com banco de dados e infraestrutura (deploy, backups). Implementa e revisa funcionalidades backend (APIs, armazenamento, endpoints de download).
- **Frontend / UX & Acessibilidade — Pedro Lucas**
Desenvolve interfaces, protótipos e implementa padrões de usabilidade e acessibilidade focados no público idoso. Responsável por validação de usabilidade em testes com usuários.
- **QA / Documentação / Integração & Suporte — Bruno Meleiro**
Escreve e executa testes, cuida da documentação (README, manual do administrador), coordena builds/CI e dá suporte a deploys e verificação pós-deploy. Auxilia no desenvolvimento quando necessário.

Ferramentas de apoio

- Git + GitHub (repositório único).
- Kanban board (GitHub Projects) com colunas: Backlog / Ready / In Progress / Review / Staging / Done.
- CI: GitHub Actions (build, lint, testes).
Banco de dados: PostgreSQL (via Docker para dev).
- Testes de acessibilidade: axe-core (integração automática em CI) + checklist manual.
- Monitoramento simples: logs e alertas básicos (Sentry ou alternativa leve).

Fluxo de trabalho (pipeline de entrega)

1. **Criar Issue** — qualquer novo requisito/bug entra como Issue com vínculo ao RF (ex.: RF7) e critérios de aceitação (mínimo: contexto + ação + resultado esperado).

2. **Revisão rápida (DoR)** — PO valida se a Issue tem descrição e critérios mínimos; então marca Ready.
3. **Desenvolvimento** — branch curto: `feature/RFx-descricao` ou `bugfix/ID`. Branches devem durar no máximo 3 dias ativos.
4. **PR + CI** — ao abrir PR, CI roda build + lint + testes; ao menos 1 revisão técnica aprova o PR. Checklist DoD deve ser preenchido no PR.
5. **Staging** — mesclado em `main` deploya em staging automaticamente; QA executa testes manuais e acessibilidade.
6. **Validação PO** — PO testa os critérios de aceitação; quando aceito, planeja-se release para produção.
7. **Produção** — deploy e monitoramento; hotfixes para problemas críticos.

Cadência de reuniões e cerimônias

- **Planejamento semanal (45 min):** PO + equipe. Seleção e quebra de itens para a semana.
- **Check rápido no board (2× por semana, 10–15 min):** atualizações rápidas — substitui daily formal.
- **Demo quinzenal (30–45 min):** apresentar incrementos ao PO e stakeholders; colher feedback.
- **Retrospectiva mensal (30–45 min):** ajustar processo com base em métricas e impedimentos.

Controle de qualidade e testes

- **DoR (Definition of Ready):** descrição clara, critérios de aceitação, link para requisito RF, estimativa (pequena/média/grande), dependências identificadas.
- **Testes com usuários:** ao menos 3 sessões de usabilidade com representantes do público idoso nas fases de validação do MVP (rastrear problemas e ajustar).
- **DoD (Definition of Done) — checklist mínimo no PR:**
 - Build e linter passando no CI
 - Testes unitários/integração relevantes passando
 - Revisão de código aprovada por ≥ 1 colega
 - Testes básicos de acessibilidade (axe-core) executados
 - Documentação (README / changelog) atualizada
 - Deployed em staging para verificação do PO

Definições formais

- **Definition of Ready (DoR)** — critério mínimo para que uma issue entre em desenvolvimento (descrição clara, critérios de aceitação, vínculo a requisito RF, estimativa, dependências identificadas).
- **Definition of Done (DoD)** — checklist de aceite para cada PR (código compilando, testes automatizados passando, revisões feitas, documentação atualizada, testes de acessibilidade executados, deploy em staging validado pelo PO).

Rastreabilidade e convenções mínimas

- Labels GitHub: RF1, RF2, bug, enhancement, accessibility.
- Branches: feature/RF7, nome, bugfix ISSUE-123, hotfix/descricao.
- Cada Issue/PR referencia os requisito(s) do Capítulo 4 para garantir rastreabilidade.
- Backups: backups diários do banco; testados em staging ao menos 1 vez antes da entrega final.

Métricas simples a acompanhar

- **Lead time** (criação → produção), **cycle time** (início → merge), **throughput** (itens concluídos/semana). Registrar impedimentos principais para priorização.

3.3 Cronograma previsto

Sprint 1 — Semanas 1–2: Setup e protótipo funcional

- Atividades: criação do repositório, configuração de CI, Docker (ambientes), definição de roles, protótipo de baixa fidelidade (wireframes) e mapeamento dos RFs prioritários (upload, busca, download, visualização).
- Responsáveis: Carlos (PO) define prioridades; Thiago cria infra e CI; Pedro desenha wireframes; Bruno prepara template de Issue/DoD.
- Entregável: repo com CI funcional + wireframes aprovados pelo PO.
- CA: CI roda com sucesso; wireframes validados pelo PO.

Sprint 2 — Semanas 3–4: MVP — Backend básico + estrutura de dados

- Atividades: modelagem DB, endpoints básicos (upload meta, listagem, download), armazenagem de arquivos (local para dev), API de busca simples (texto).
- Responsáveis: Thiago (lead backend) + Bruno (integração CI/testes). Pedro apoia testes e documentação do endpoint.
- Entregável: API funcional em staging com upload e download básicos.
- CA: upload com metadados, download funcionando, testes unitários básicos.

Sprint 3 — Semanas 5–6: MVP — Frontend inicial e integração

- Atividades: páginas principais (home, busca, detalhe do item, botão de download), integração com API, preview de arquivos (quando aplicável).
- Responsáveis: Pedro (frontend), Thiago (API fixes), Bruno (testes). Carlos faz validação com stakeholders em fim de sprint.
- Entregável: protótipo hi-fi implementado e integrado no staging.
- CA: usuário externo consegue buscar, visualizar e baixar um REA.

Sprint 4 — Semanas 7–8: Autenticação administrativa e upload avançado

- Atividades: área administrativa básica (login seguro), upload com validação de metadados obrigatórios e associação de licença; logs de upload.
- Responsáveis: Thiago (autenticação), Pedro (UI admin), Bruno (testes/login). Carlos aprova fluxo admin.
- Entregável: painel administrativo mínimo com upload e gerenciamento de itens.
- CA: administrador cadastrado consegue logar, enviar e editar metadados.

Sprint 5 — Semanas 9–10: Acessibilidade, usabilidade e testes com usuários

- Atividades: ajustes de acessibilidade (WCAG AA básico), testes com pelo menos 3 usuários idosos, correções de usabilidade, integração de axe-core no CI.
- Responsáveis: Pedro lidera testes de usabilidade; Bruno executa integração axe-core; Thiago corrige backend quando necessário. Carlos organiza sessões com stakeholders/usuários.
- Entregável: relatório de usabilidade + correções aplicadas.
- CA: resultados de testes incorporados; axe-core sem bloqueadores críticos.

Sprint 6 — Semanas 11–12: Documentação e preparação para entrega

- Atividades: documentação de uso (manual do administrador e guia rápido para usuários), políticas de backup e recuperação testadas em staging, checklist final de DoD para MVP.
- Responsáveis: Bruno (docs e backups), Pedro (ajustes UI), Thiago (scripts de backup). Carlos revisa a documentação.
Entregável: documentação completa + teste de restauração do backup realizado com sucesso.
- CA: documentação aprovada pelo PO; restauração testada.

Sprint 7 — Semanas 13–14: Polimento, correções finais e apresentação

- Atividades: correções finais, preparar apresentação/demo final, tempo reservado para bugs críticos e benchmarking de performance leve.
- Responsáveis: todos (pairing para correções).
- Entregável: versão de entrega + demo final para stakeholders e arquivo de release (changelog, RFs atendidos).
- CA: demo aceita pelo PO; checklist DoD global atendido; plano de manutenção definido.

4 Requisitos do Sistema

4.1 Requisitos Funcionais

ID	Funcionalidade	Prioridade
RF1	O sistema deve conter uma barra de pesquisa para os materiais (REAs).	Essencial
RF2	O sistema deve salvar os metadados dos materiais recebidos.	Essencial
RF3	O sistema deve ser capaz de oferecer uma pesquisa por metadados.	Importante
RF4	O sistema deve ser capaz de receber arquivos (de administradores)	Essencial
RF5	O sistema deve conter autenticação (a princípio somente de admin para o TEDI)	Essencial
RF6	O sistema deve permitir que conteúdos (REAs) sejam armazenados, editados e recuperados de forma persistente.	Essencial
RF7	O sistema deve oferecer a opção de baixar os REAs	Essencial
RF8	Cada material tem sua página individual com título, descrição e botões de visualizar e baixar o conteúdo	Essencial

4.2 Requisitos Não-funcionais

ID	Requisito	Categoria
RN1	O sistema deve estar em conformidade com a lei 14.198 ¹	Legislação
RN2	O sistema deve estar disponível apenas no ambiente web	Compatibilidade
RN3	A interface deve ser acessível para idosos	Usabilidade
RN4	Todas as ferramentas usadas no desenvolvimento devem ser gratuitas	Recursos
RN5	O sistema dependerá de conexão à internet para ser acessado	Disponibilidade

4.3 Diagrama de Casos de Uso

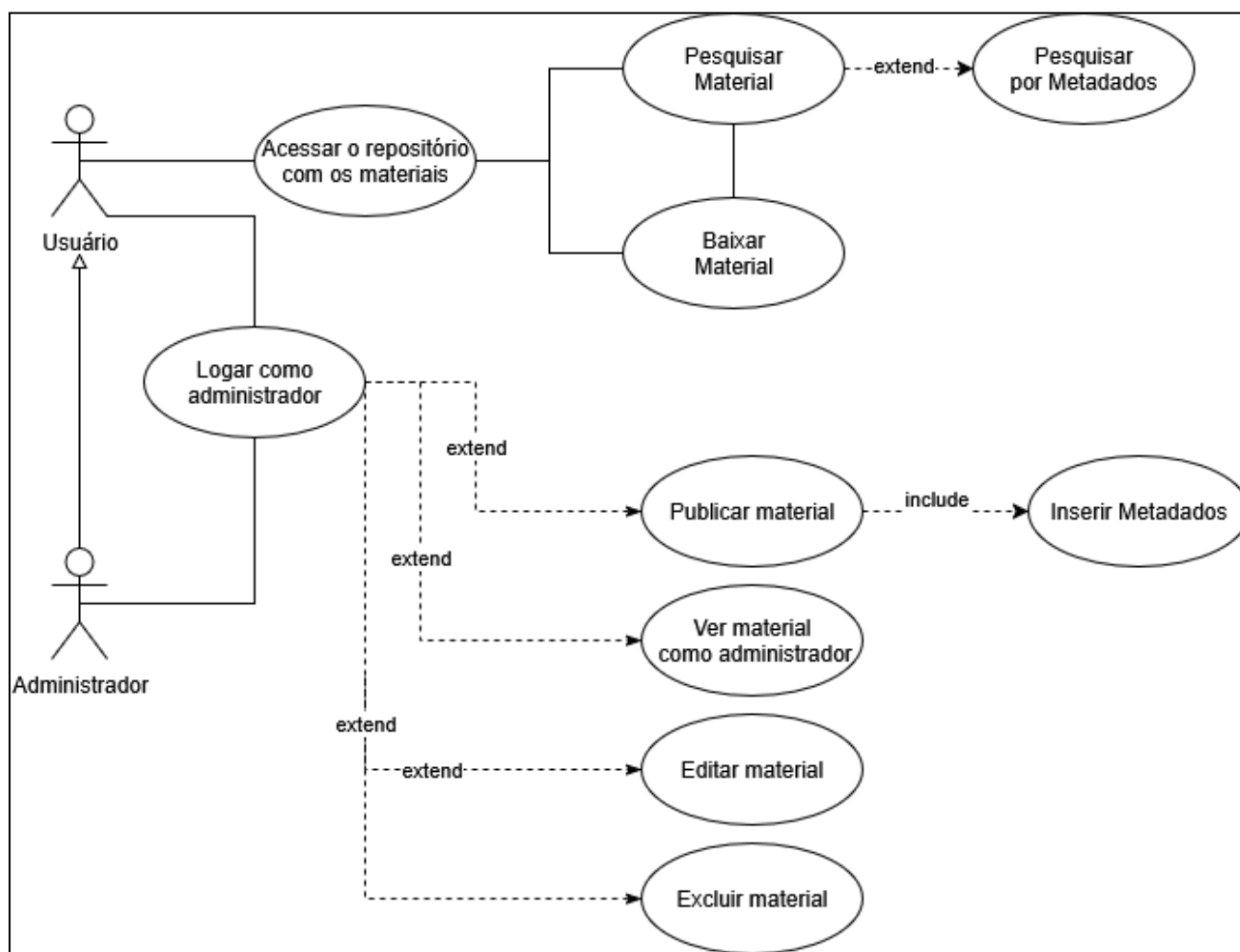


Figura 1: Visão Geral do Sistema (Login [RF5], Baixar [RF7], Pesquisa[RF3])

O diagrama de caso de uso (Figura 1) exemplifica o fluxo que vai desde o login (RF5) por parte dos administradores, até a manipulação dos materiais (como publicação, edição, exclusão e visualização de administrador). Além da possibilidade de acesso aos conteúdos e download dos materiais por parte do usuário, contando também com pesquisas especificando, ou não, metadados (como autor, data de publicação, etc.).

4.4 Protótipos de Telas

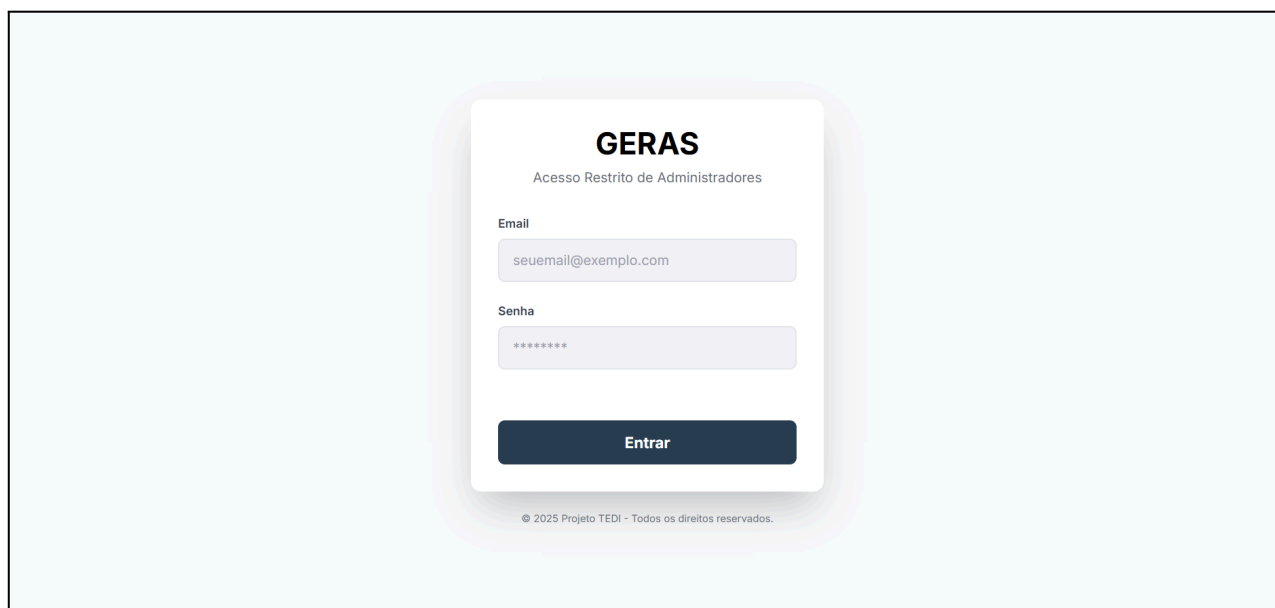
Métricas simples a acompanhar

- **Design legível:** fonte base grande (mínimo 16–18 px para texto corpo), espaçamento generoso, botões grandes (mín. 44×44 px) e áreas de clique/touch ampliadas.
- **Contraste alto:** relação de contraste que atenda WCAG AA ou superior para texto e elementos interativos.
- **Navegação simples:** menu reduzido (Home, Busca, Sobre, Acessibilidade, Área Administrativa) e caminho claro para ações principais (buscar, visualizar, baixar).
- **Feedback claro:** mensagens de sucesso/erro visíveis e persistentes; indicadores de carregamento durante uploads/consultas.
- **Acessibilidade:** marcação semântica (ARIA), suporte a navegação por teclado, labels explícitos para leitores de tela e descrição alternativa para imagens.
- **Microcopy amigável:** linguagem simples, instruções curtas e exemplos explicativos junto aos campos (ex.: “Adicione um resumo de 1–2 frases”).
- **Modo de alto contraste / tamanho de fonte ajustável:** controle visível na interface (botão “A+ / A-” e alternância de contraste).



Figura 2: Página Inicial

A página inicial (Figura 2) é o coração de tudo, com uma breve descrição do que são os REAs, uma barra de pesquisa e alguns exemplos de materiais, ela garante um contato com grande parte do sistema de maneira muito facilitada. Sendo essa porta de entrada, ela permite acessar os “resultados da pesquisa” (ainda sem protótipo), também permite acessar as “páginas de material” (figura 5) dos materiais destacados e as páginas do menu, sendo elas “login” (figura 3), “sobre” (ainda sem protótipo), “todos os materiais”(ainda sem protótipo), e a própria página inicial.



The image shows a login page for a system named GERAS. The page has a light blue background. In the center, there is a white card with rounded corners and a subtle shadow. The card contains the following elements: the title 'GERAS' in bold black text, the subtitle 'Acesso Restrito de Administradores' in a smaller font, a label 'Email' above a text input field containing 'seuemail@exemplo.com', a label 'Senha' above a password input field with masked characters '*****', and a dark blue button labeled 'Entrar'. At the bottom of the card, there is a small copyright notice: '© 2025 Projeto TEDI - Todos os direitos reservados.'

Figura 3: Página de Login

A página de login (Figura 3) é a chave de acesso para a área de “inserção de material” (figura 5), sendo a única forma de inserir algum conteúdo. Nela o usuário deve informar o email e senha para poder realizar a autenticação.

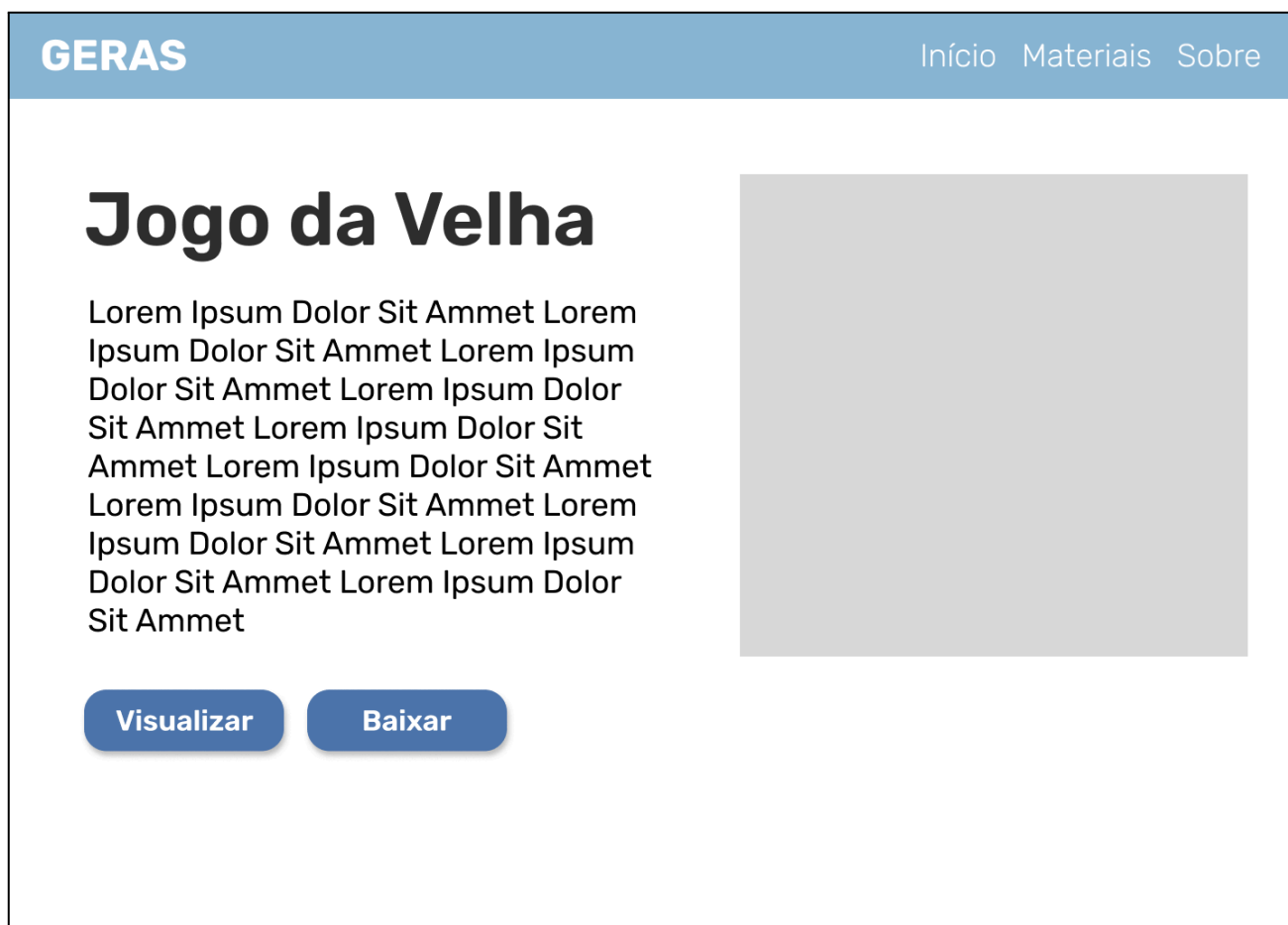
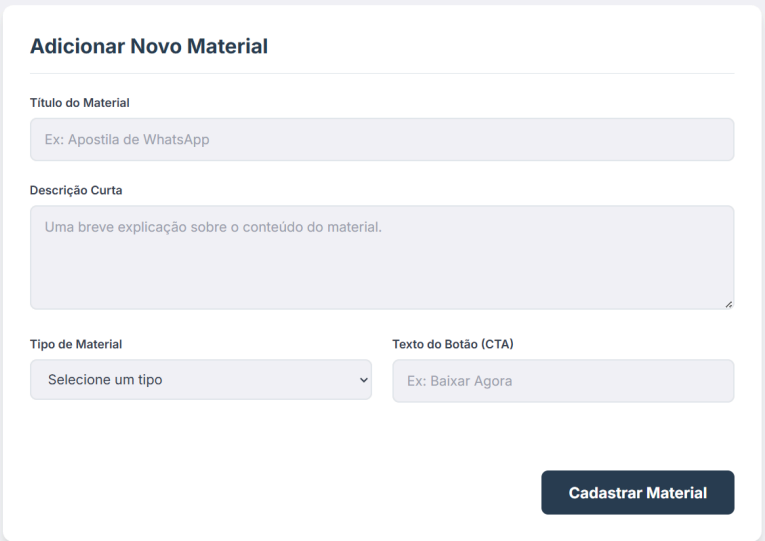


Figura 4: Página de Material

A figura 4 apresenta a página do material, cada material tem sua página individual, com título, autor, descrição, data e tipo de material. Estas páginas podem ser acessadas por pesquisas de conteúdo, pela própria "página inicial, ou ainda localizando elas em "todos os materiais". Elas por sua vez permitem visualizar e baixar o material além de darem acesso a todas as páginas do menu.



O formulário, intitulado "Adicionar Novo Material", está centralizado em uma caixa branca sobre um fundo cinza claro. Ele contém os seguintes campos: um campo de texto para o "Título do Material" com o exemplo "Ex: Apostila de WhatsApp"; um campo de texto maior para a "Descrição Curta" com o exemplo "Uma breve explicação sobre o conteúdo do material."; um menu suspenso para o "Tipo de Material" com a opção "Selecione um tipo"; e um campo de texto para o "Texto do Botão (CTA)" com o exemplo "Ex: Baixar Agora". Um botão azul escuro com o texto "Cadastrar Material" está posicionado no canto inferior direito do formulário.

Figura 5: Inserção de Material

A Figura 5, mostra a página com o formulário de inserção de materiais, serve para alimentar o sistema com novos conteúdos, contando com título, descrição e tipo de arquivo. Esta página tem acesso à página inicial (figura 2) e ao login novamente, por meio da opção “sair” (oculta na captura da figura).

5 Análise do Sistema

5.1 Modelo de entidade do Banco de Dados

O modelo de entidade e relacionamento do banco de dados, Em que a tabela principal é a tabela de recursosEducaionais, onde é armazenado as as atividades que estarão disponíveis para acesso no sistema, contando com algumas tabelas de apoio e indexação que torna a pesquisa mais rápida, como as tabelas palavraChave, recursoPalavraChave e Categoria.

O modelo também leva em consideração que no futuro o sistema pode ser utilizado por outras instituições e/ou grupos de uma mesma instituição, então possui tabelas individuais de instituições e grupos.

Já a tabela de usuários é para fazer o controle de usuários normais, administradores e colaboradores de forma mais prática.

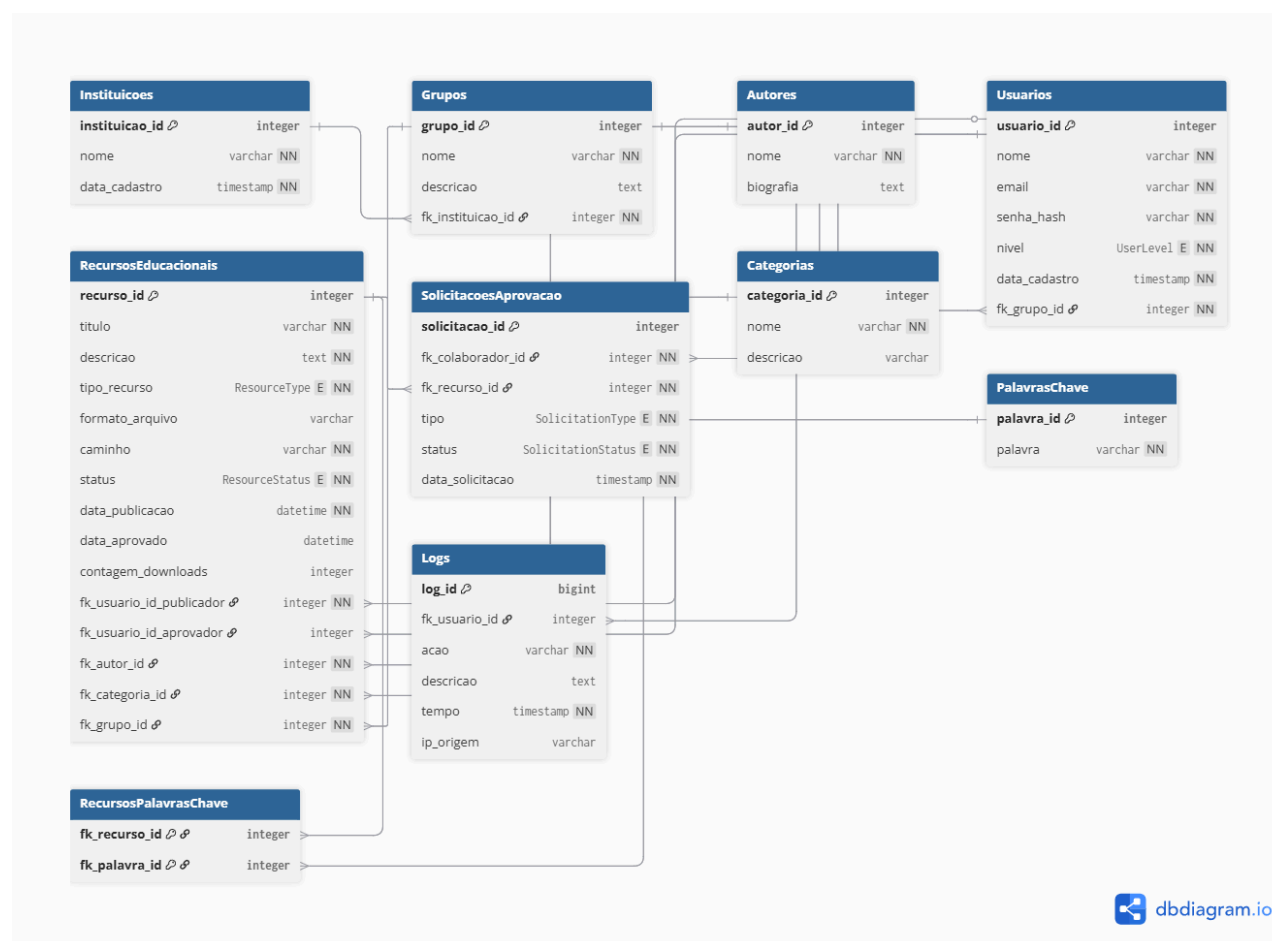


Figura 6: Diagrama de entidade e relacionamento do banco de dados

5.2 Dicionário de dados:

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
instituicoes	instituicao_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único da instituição.
Descrição da tabela:	nome	varchar(255)	Não Nulo	Nome oficial da instituição.
- Armazena as instituições que utiliza o sistema	data_cadastro	timestamp	Não Nulo	Data e hora em que a instituição foi cadastrada.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
grupos	grupo_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único do grupo.
Descrição da tabela:	nome	varchar(255)	- Não Nulo	Nome do grupo.
- Armazena os grupos que pertencem a qual instituição	descricao	text		Descrição opcional sobre os objetivos ou membros do grupo.
	fk_instituicao_id	integer	Não Nulo, Chave Estrangeira (FK)	Referencia a instituição (instituicoes.instituicao_id) à qual o grupo pertence.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
autores	autor_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único do autor.
Descrição da tabela:	nome	varchar(255)	Não Nulo	Nome completo do autor.
- Armazena o nome do autor de algum recurso educacional	biografia	text	Permite Nulo	Biografia ou informações adicionais sobre o autor.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
log	log_id	bigint	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único do registro de log.
Descrição da tabela:	fk_usuario_id	integer	Permite Nulo, Chave Estrangeira (FK)	Referencia o usuário (usuarios.usuario_id) que realizou a ação.
- Armazena um log das ações realizadas no banco.	ação	varchar(100)	Não Nulo	Descrição curta da ação (ex: 'LOGIN', 'DOWNLOAD_RECURSO').
	descricao	text	Permite Nulo	Detalhes adicionais sobre a ação registrada.
	tempo	timestamp	Não Nulo	Data e hora exatas em que a ação ocorreu.
	ip_origem	varchar(45)	Permite Nulo	Endereço IP de onde a requisição se originou.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
palavraChave	palavra_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único da palavra-chave.
Descrição da tabela:	palavra	varchar(50)	Não Nulo, Único	A palavra-chave em si.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
categorias	categoria_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único da categoria.
Descrição da tabela:	nome	varchar(100)	Não Nulo	Nome da categoria.
- Armazena as categorias dos recursos para pesquisa	descricao	varchar(255)	Permite Nulo	Descrição opcional da categoria.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
autores	autor_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único do autor.
Descrição da tabela:	nome	varchar(255)	Não Nulo	Nome completo do autor.
- Armazena o nome do autor de algum recurso educacional	biografia	text	Permite Nulo	Biografia ou informações adicionais sobre o autor.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
usuarios	usuario_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único do usuário.
Descrição da tabela:	nome	varchar(255)	Não Nulo	Nome completo do usuário.
Armazena os dados dos usuario	email	varchar(255)	Não Nulo, Único	Endereço de e-mail do usuário, usado para login.
	senha_hash	varchar(255)	Não Nulo	Hash da senha do usuário para armazenamento seguro.
	nivel	enum	Não Nulo	Nível de permissão: administrador ou colaborador.
	data_cadastro	timestamp	Não Nulo	Data e hora em que o usuário foi cadastrado.
	fk_grupo_id	integer	Não Nulo, Chave Estrangeira (FK)	Referencia o grupo (grupos.grupo_id) ao qual o usuário pertence.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
recursosEducativos	recurso_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador único do recurso.
Descrição da tabela:	titulo	varchar(255)	Não Nulo	Título do recurso educacional.
- Armazena os dados dos recursos educativos	descricao	text	Não Nulo	Descrição detalhada do conteúdo do recurso.
- de maneira local ou por forma de link	tipo_recurso	enum	Não Nulo	Tipo entre: 'arquivo' e 'link_externo'.
	formato_arquivo	varchar(10)	Não Nulo	Formato do arquivo (ex: 'PDF', 'DOCX', 'MP4').
	caminho	varchar(255)	Não Nulo	Caminho para o arquivo no servidor ou a URL do link externo.
	status	enum	Não Nulo	Status de moderação: aprovado, pendente ou rejeitado.
	data_publicacao	datetime	Não Nulo	Data e hora em que o recurso foi submetido.
	data_aprovado	datetime	Permite Nulo	Data e hora da aprovação do recurso.
	contagem_downloads	integer	Não Nulo	Contador de downloads do recurso.
	fk_usuario_id_publicador	integer	Não Nulo, Chave Estrangeira (FK)	Referencia o usuário (usuarios.usuario_id) que publicou o recurso.
	fk_usuario_id_aprovador	integer	Permite Nulo, Chave Estrangeira (FK)	Referencia o usuário (usuarios.usuario_id) que aprovou o recurso.
	fk_autor_id	integer	Não Nulo, Chave Estrangeira (FK)	Referencia o autor (autores.autor_id) do conteúdo.
	fk_categoria_id	integer	Não Nulo, Chave Estrangeira (FK)	Referencia a categoria (categorias.categoria_id) do recurso.
	fk_grupo_id	integer	Não Nulo, Chave Estrangeira (FK)	Referencia o grupo que possui o recurso.

Tabela	Campo	Tipo de Dado	Restrições/Regras	Descrição
SolicitacoesAprovacao	solicitacao_id	integer	Chave Primária (PK), Não Nulo, Autoincremento	Identificador unico da solicitação.
Descrição da tabela:	fk_colaborador_id	integer	Não Nulo	Referencia o ID do colaborador.
Armazena as solicitações de alteração no sistema realizada por	fk_recurso_id	integer	Não Nulo	Referencia o ID do recurso a ser alterado.
Colaboradores	tipo	enum	Não Nulo	Tipo de solicitação, entre: Edicao e Exclusao.
	status	enum	Não Nulo	Status da solitação, entre: Pendente, Aprovada, Rejeitada.
	data_solicitacao	timestamp	Não Nulo	Data da solicitação.

5.4 Diagramas de Atividades:

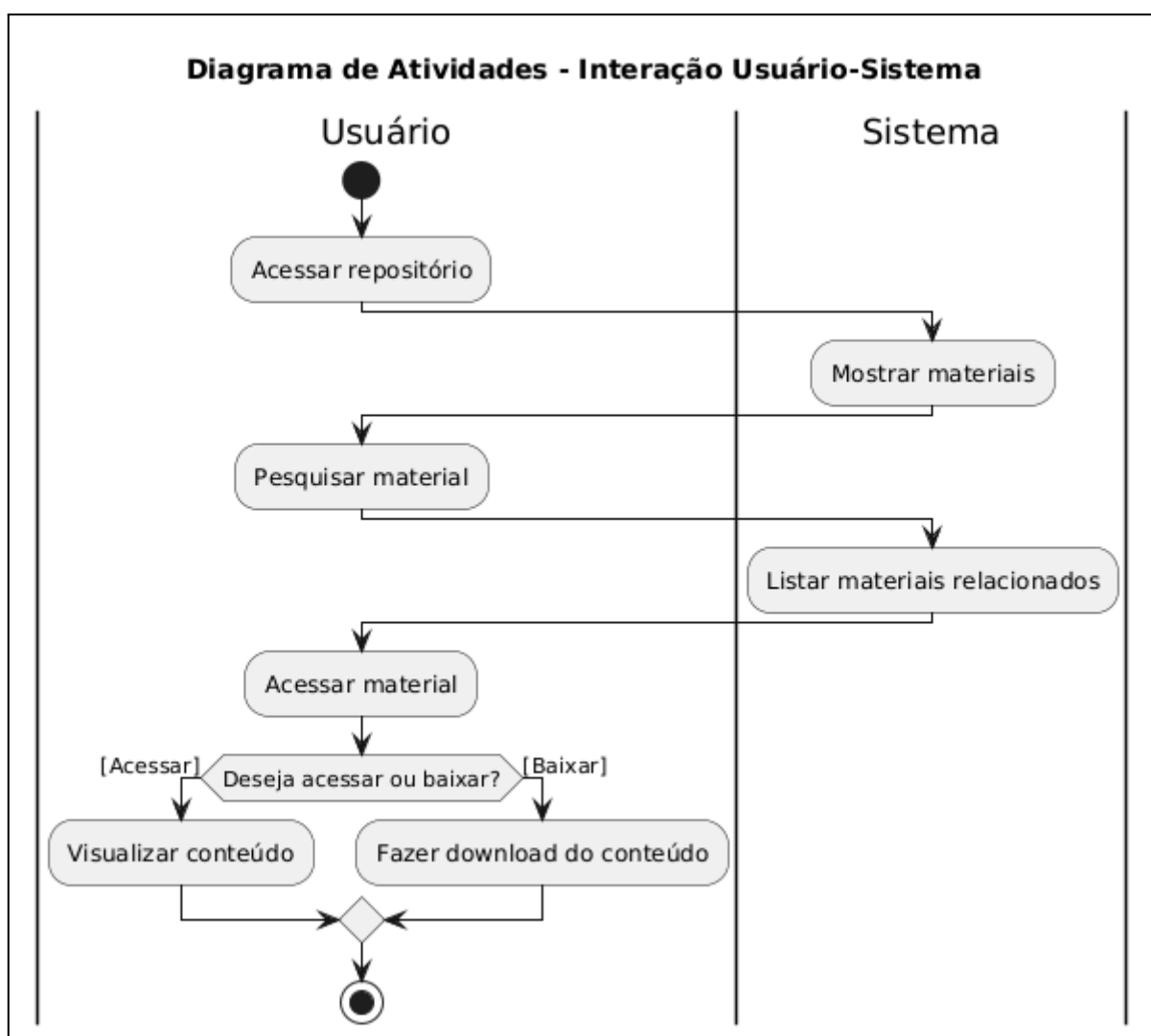


Figura 8: Diagrama de Atividades Usuário-Sistema

Referente ao diagrama de atividades representando as interações usuário-sistema (figura 8), o principal fluxo gira em torno de encontrar o material desejado (em geral, por meio de pesquisa), com possibilidade de acesso direto ou download do conteúdo por meio do repositório.

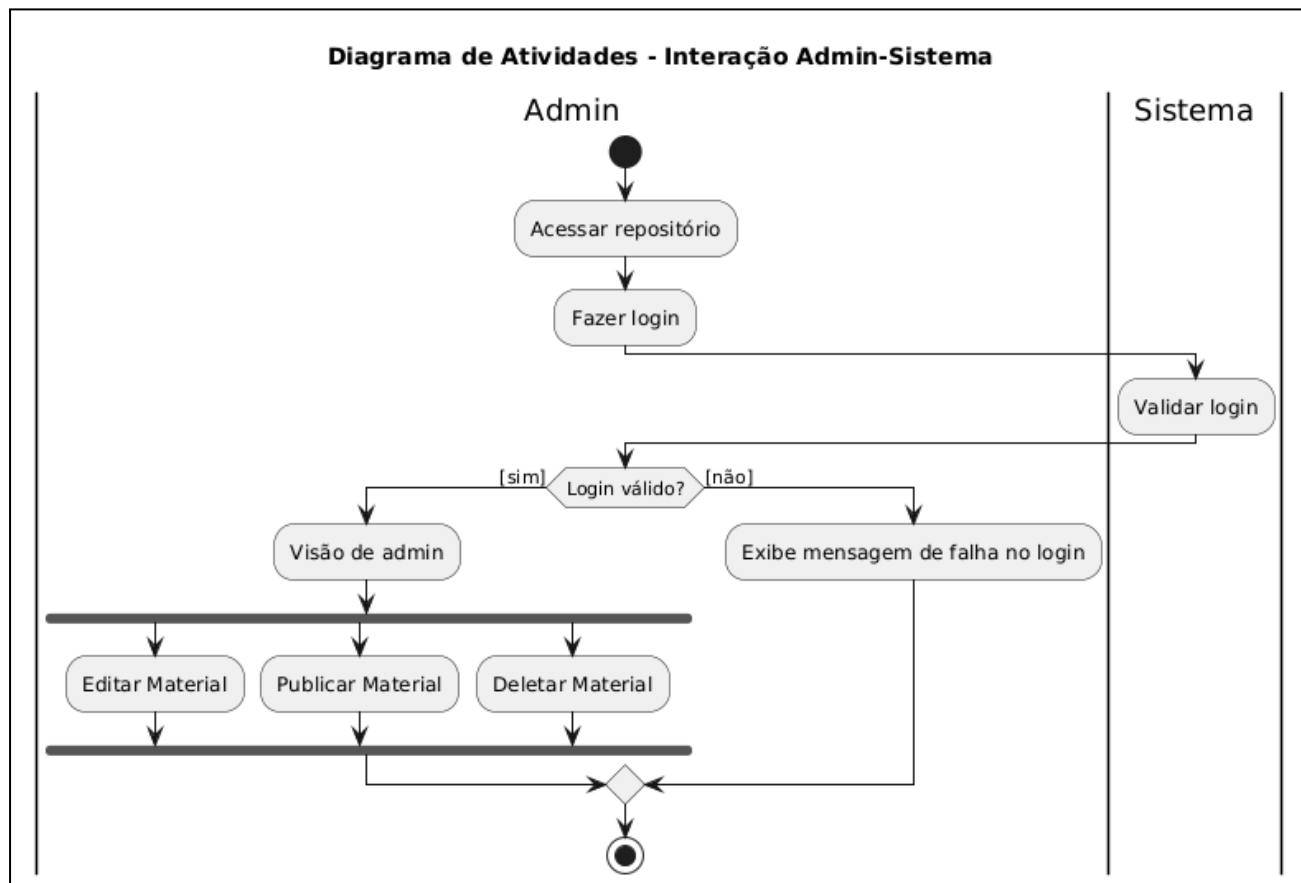


Figura 9: Diagrama de Atividades Admin-Sistema

A relação entre administrador e sistema (Figura 9) é constituída principalmente de login e operações básicas, girando em torno de publicar, editar e deletar os conteúdos.

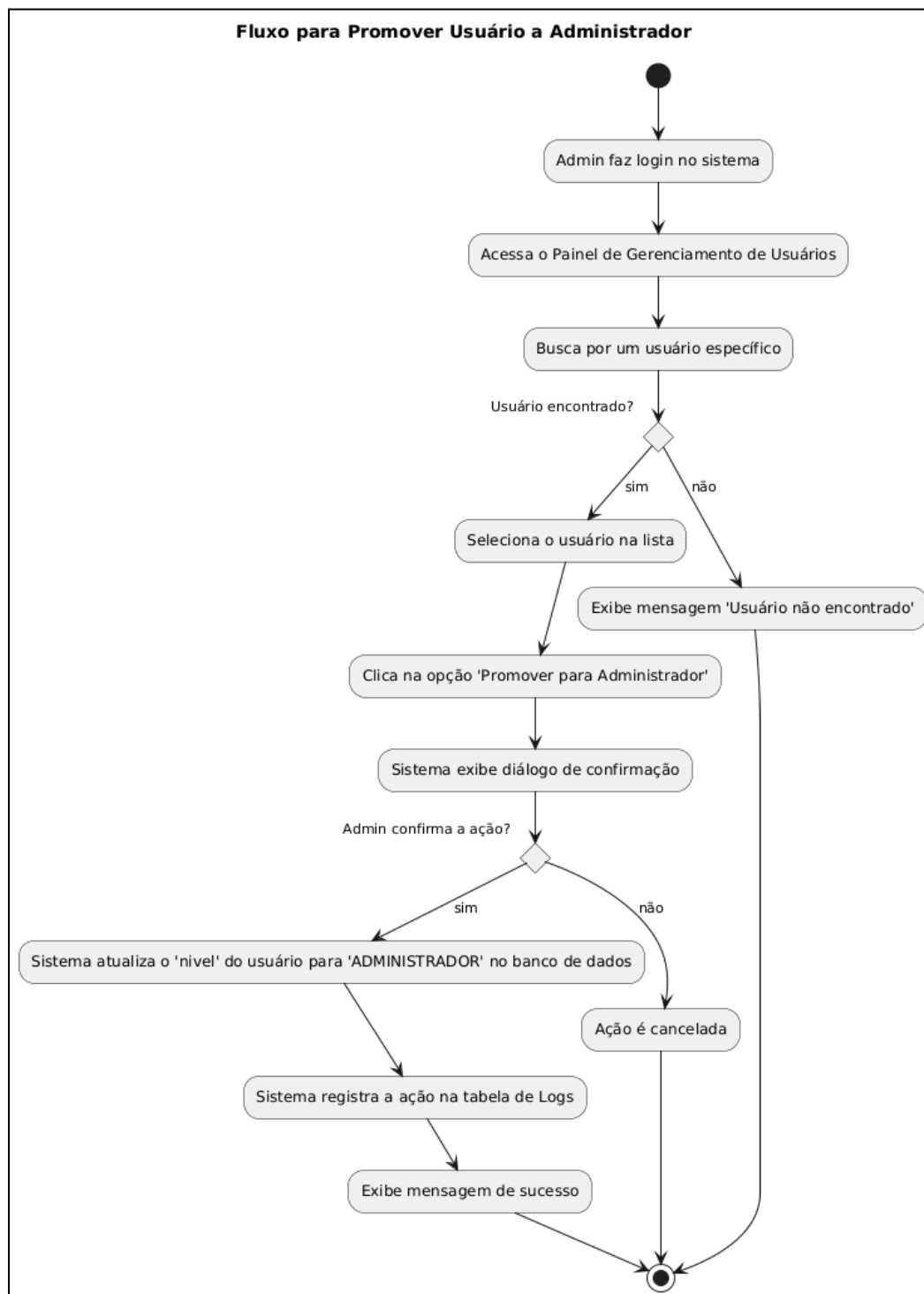


Figura 10: Diagrama de Atividades Admin-Sistema

A relação entre administrador e usuário (Figura 10), onde é demonstrado o processo de transformação de um usuário normal para um usuário do tipo colaborador.

6 Bibliografia

¹ UNESCO. **Open Educational Resources**. Disponível em:

<https://www.unesco.org/en/open-educational-resources>. Acesso em: 8 out. 2025.

² **Lei nº 14.198**. Disponível em:

https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2021/lei/l14198.htm Acesso em: 10 out. 2025.