

ETEC JARDIM ANGELA
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

ANNA BEATRIZ OLIMPIO DE SOUSA

GUSTAVO OLIVEIRA DA SILVA

IVAN APARECIDO NASCIMENTO DE JESUS

MURILO COSTA DE SANTANA

RENAN GABRIEL CAMARGO OLIVEIRA SILVA

KAIROS
SISTEMA DE GESTÃO DE BIBLIOTECAS ONLINE

SÃO PAULO

2026

ANNA BEATRIZ OLIMPIO DE SOUSA

GUSTAVO OLIVEIRA DA SILVA

IVAN APARECIDO NASCIMENTO DE JESUS

MURILO COSTA DE SANTANA

RENAN GABRIEL CAMARGO OLIVEIRA SILVA

Trabalho de Conclusão de Curso
apresentado ao Centro Estadual de
Educação Tecnológica Paula Souza, como
requisito para a obtenção do diploma de
Técnico em Desenvolvimento de Sistemas
sob a orientação do Professor Especialista
Ericles Araujo.

SÃO PAULO

2026

SUMÁRIO

1.	INTRODUÇÃO	7
2.	JUSTIFICATIVA	8
3.	PROBLEMA	9
4.	HIPÓTESE.....	10
5.	OBJETIVOS	11
5.1.	OBJETIVO GERAL	11
5.2.	OBJETIVOS ESPECÍFICOS.....	11
6.	REQUISITOS FUNCIONAIS.....	12
7.	REQUISITOS NÃO FUNCIONAIS	14
8.	METODOLOGIA DA PESQUISA.....	15
9.	PERGUNTAS E RESPOSTAS.....	16
10.	TRABALHOS RELACIONADOS.....	17
11.	METODOLOGIA DO TRABALHO	19
12.	FERRAMENTAS E LINGUAGENS UTILIZADAS	20
13.	CRONOGRAMA DO TRABALHO	22
14.	DOCUMENTAÇÃO DO SOFTWARE.....	23
14.1.	FLUXOGRAMA	23
15.	DIAGRAMA DE CASO DE USO	24
16.	MODELO CONCEITUAL DO BANCO DE DADOS.....	25
17.	MODELO LÓGICO DO BANCO DE DADOS	26
18.	ARQUITETURA DO SISTEMA	27
19.	TELAS DO SISTEMA.....	31
20.	CÓDIGOS-FONTE DO SISTEMA	34
21.	CICLO DE VIDA DO PROGRAMA.....	39
21.1.	SOLICITAÇÕES DE EMPRÉSTIMO	39
21.2.	NOTIFICAÇÕES.....	39
21.3.	CADASTRO COM VALIDAÇÃO DE CPF, EMAIL E SENHA – HASH COM BSCRYPT.....	39
22.	CONSIDERAÇÕES FINAIS.....	40
23.	REFERÊNCIAS.....	41

LISTA DE FIGURAS

Figura 1 - Cronograma de atividades do projeto, distribuído por bimestre, com as etapas de levantamento de requisitos, desenvolvimento, testes e documentação. Página 20.

Figura 2 - Fluxograma do sistema de gestão de biblioteca, representando o fluxo de ações do usuário e do administrador, desde o acesso ao catálogo até a conclusão do empréstimo e devolução. Página 21.

Figura 3 - Diagrama de caso de uso do sistema, ilustrando os atores (Usuário Leitor e Administrador) e as funcionalidades disponíveis para cada perfil, como solicitar empréstimo, devolver livro, gerenciar acervo e aprovar solicitações. Página 23.

Figura 4 - Modelo Conceitual do banco de dados, representando as entidades principais do sistema e seus relacionamentos em alto nível, sem especificação de tipos de dados ou chaves. Página 24.

Figura 5 - Modelo Lógico do banco de dados, apresentando as tabelas, atributos, chaves primárias e estrangeiras que estruturam o armazenamento das informações do sistema. Página 25.

Figura 6 - Diagrama de arquitetura geral do sistema, evidenciando a separação entre front-end, back-end e banco de dados e a comunicação entre as camadas via requisições HTTP. Página 26.

Figura 7 - Estrutura de pastas do projeto, com a organização dos diretórios de front-end e back-end. Página 27.

Figura 8 - Diagrama de fluxo de requisição e resposta entre o cliente e o servidor Node.js, passando pelo roteamento Express e a camada de acesso ao banco MySQL. Página 28.

Figura 9 - Código que valida a mensagem de atraso e o prazo de 14 dias para a devolução. Página 28.

Figura 10 - Tela de login do sistema, com campos de e-mail e senha e acesso diferenciado para usuários comuns e administradores. Página 30.

Figura 11 - Tela de catálogo de livros, exibindo o acervo disponível com opção de busca e botão para solicitação de empréstimo. Página 30.

Figura 12 - Tela de gerenciamento de solicitações de empréstimo, acessível ao administrador, com opções de aprovação ou reprovação de cada pedido. Página 31.

Figura 13 - Tela de gerenciamento de empréstimos ativos, com informações de prazo, status e botão para registro de devolução. Página 32.

Figura 14 - Código-fonte do back-end: rota de login de usuário com validação de credenciais e comparação de senha via bcrypt. Página 33.

Figura 15 - Código-fonte do back-end: rota de solicitação de empréstimo, com verificações de duplicidade, disponibilidade de exemplar e notificação aos administradores. Página 34.

Figura 16 - Código-fonte do back-end: rota de aprovação de solicitação, com criação do empréstimo, atualização do exemplar e notificação ao usuário. Página 35.

Figura 17 - Código-fonte do back-end: rota de devolução de livro, com atualização do status do empréstimo, liberação do exemplar e geração de notificação de avaliação. Página 35.

Figura 18 - Código-fonte do front-end: funções JavaScript responsáveis pelo envio das requisições de login, solicitação de empréstimo e registro de devolução. Página 36.

Figura 19 - Código que representa a regra dos 14 dias a partir do atraso na devolução do livro. Página 37.

1. INTRODUÇÃO

Com o avanço da tecnologia e a crescente necessidade de organização e acesso eficiente à informação, torna-se fundamental a modernização dos sistemas utilizados em ambientes educacionais, especialmente nas bibliotecas escolares.

Nesse contexto, o presente projeto propõe o desenvolvimento de um sistema digital voltado ao gerenciamento de acervos bibliográficos, com foco inicial na ETEC Jardim Ângela.

Atualmente, muitas instituições ainda utilizam métodos tradicionais, como planilhas eletrônicas e registros manuais, que apresentam limitações quanto à organização, controle e confiabilidade das informações.

Esses fatores impactam diretamente na eficiência dos processos de empréstimo, devolução e localização de livros, além de dificultarem a gestão adequada do acervo.

Diante desse cenário, o projeto busca oferecer uma solução tecnológica capaz de otimizar o funcionamento da biblioteca, promovendo maior agilidade, segurança e precisão no controle das informações.

Além disso, a proposta visa contribuir para a democratização do acesso à leitura e à informação, especialmente em regiões com maior vulnerabilidade social, fortalecendo o papel das bibliotecas como espaços de conhecimento e inclusão.

Por meio da automatização dos processos e da utilização de ferramentas digitais, espera-se não apenas melhorar a gestão interna da biblioteca, mas também proporcionar uma experiência mais eficiente e acessível para alunos e profissionais envolvidos.

2. JUSTIFICATIVA

O presente projeto propõe o desenvolvimento de um sistema voltado ao gerenciamento de livros da ETEC Jardim Ângela, motivado pela observação das dificuldades enfrentadas no modelo atual de controle.

A iniciativa busca otimizar a administração dos acervos e tem potencial de expansão para outras instituições da cidade de São Paulo.

A adesão do sistema poderá contribuir significativamente para o aprimoramento do funcionamento de bibliotecas escolares, oferecendo uma ferramenta mais eficiente e acessível, especialmente em regiões com alta densidade populacional.

Além disso, o projeto visa reduzir desigualdades entre colégios públicos e particulares, promovendo o acesso equitativo à literatura e aos recursos tecnológicos.

Em nível microrregional, como no bairro Jardim São Luís, onde há instituições públicas e particulares com diferentes condições de infraestrutura, a implementação do sistema poderia equilibrar o acesso à gestão bibliotecária, favorecendo escolas públicas e fortalecendo o incentivo à leitura em áreas de maior vulnerabilidade social.

3. PROBLEMA

Observa-se uma significativa precariedade no controle das informações da biblioteca de diversas escolas, uma vez que a instituição ainda utiliza ferramentas como o Excel e registros manuscritos em papel, métodos que se mostram pouco eficazes pela complexidade da tarefa.

Essas práticas dificultam a organização, o acesso e a confiabilidade dos dados relacionados ao acervo.

A escolha do tema deste projeto surgiu a partir das dificuldades identificadas no gerenciamento dos processos de empréstimo e devolução de livros, especialmente devido ao uso do Excel, que não é uma ferramenta ideal para esse tipo de controle, tornando as atividades mais complexas e suscetíveis a erros.

Durante uma pesquisa de campo realizada pelos integrantes do grupo em outubro de 2025, no Colégio Carolina Sintra, localizado na região do Capão Redondo, foram identificados diversos problemas estruturais, dentre os quais se destacam:

- Ausência de organização adequada do acervo, com livros classificados em gêneros incorretos.
- Aplicação de restrição de idade baseada apenas no julgamento da professora, sem critérios padronizados.
- Falta de controle eficiente sobre a devolução dos livros.
- Dificuldade na identificação de qual usuário está com determinado livro, sendo necessário consultar listas extensas.
- Armazenamento inadequado de livros doados, muitos dos quais não possuem selo de autenticação governamental e ficam fora das prateleiras, disponíveis para retirada sem controle.

4. HIPÓTESE

Parte-se da hipótese de que a implementação de um sistema informatizado para o gerenciamento do acervo bibliográfico contribuirá significativamente para a melhoria dos processos internos da biblioteca.

Acredita-se que, por meio da automatização das atividades de cadastro, empréstimo e devolução de livros, será possível reduzir a incidência de falhas humanas, aumentar a precisão no controle das informações e otimizar o tempo de atendimento aos usuários.

Além disso, pressupõe-se que a utilização de uma plataforma digital especializada possibilitará maior organização do acervo, facilitando a localização dos materiais e o acompanhamento das movimentações, bem como proporcionando maior confiabilidade e segurança no armazenamento dos dados.

Dessa forma, espera-se que o sistema proposto promova maior eficiência na gestão bibliotecária e contribua para a melhoria do acesso à informação no ambiente escolar.

5. OBJETIVOS

Tem-se como objetivo desenvolver um sistema que otimize o funcionamento geral de bibliotecas escolares, melhorando o controle e a organização do acervo e aumentando a eficiência dos processos de atendimento, reduzindo falhas humanas.

5.1. OBJETIVO GERAL

Implementar um sistema digital capaz de aperfeiçoar o controle de entrada e saída dos livros, utilizando o RM (Registro de Matrícula) dos alunos para garantir maior precisão na gestão das movimentações do acervo.

5.2. OBJETIVOS ESPECÍFICOS

- Automatizar os processos de atendimento da biblioteca, por meio da implementação de ferramentas tecnológicas que reduzam a necessidade de intervenções manuais nas rotinas de empréstimo, devolução e gestão de usuários.
- Reduzir a ocorrência de falhas humanas, mediante a padronização, sistematização e digitalização dos procedimentos operacionais, assegurando maior precisão e confiabilidade nas atividades realizadas.
- Aprimorar o controle e a gestão do acervo, por meio de mecanismos que possibilitem o monitoramento detalhado da circulação de livros, da disponibilidade de exemplares e do histórico de movimentações.

6. REQUISITOS FUNCIONAIS

O sistema foi desenvolvido considerando requisitos que garantem não apenas o funcionamento adequado das funcionalidades, mas também aspectos relacionados à segurança, organização, desempenho e usabilidade.

A validação da conta por meio de código de autenticação tem como objetivo aumentar a segurança do sistema durante o processo de cadastro ou acesso do usuário. Esse mecanismo confirma a identidade do usuário por meio de um código enviado previamente, reduzindo a possibilidade de acessos não autorizados e contribuindo para a proteção das informações armazenadas.

A classificação correta dos livros por gênero literário e categoria permite uma organização eficiente do acervo, facilitando a localização e a pesquisa de obras. A categorização adequada melhora a experiência dos usuários, tornando o processo de busca mais rápido e intuitivo, além de contribuir para uma administração mais organizada da biblioteca.

O uso de identificação de dados duplicados para manter a integridade do cadastro impede o registro repetido de informações, como livros ou usuários já cadastrados. Essa validação evita inconsistências na base de dados, reduz erros administrativos e garante maior confiabilidade nas informações utilizadas pelo sistema.

A integração entre perfis de usuário e bibliotecário, com níveis de acesso distintos, estabelece diferentes permissões de acordo com as responsabilidades de cada perfil. Enquanto os usuários possuem acesso às funcionalidades relacionadas à consulta do acervo e ao gerenciamento de seus empréstimos, os bibliotecários dispõem de recursos administrativos, como cadastro, edição e exclusão de livros, além do gerenciamento de empréstimos e usuários. Essa divisão fortalece a segurança e assegura o controle adequado das operações.

A localização do exemplar no acervo, indicando disponibilidade e posição na prateleira, possibilita que usuários e bibliotecários encontrem rapidamente uma obra dentro da biblioteca. Além de informar se o livro está disponível para empréstimo, o

sistema apresenta sua localização física, contribuindo para a otimização do atendimento e da organização do acervo.

Por fim, a otimização de desempenho para funcionamento adequado em computadores mais antigos garante que o sistema possa ser utilizado em diferentes ambientes, inclusive em equipamentos com menor capacidade de processamento. Para isso, o desenvolvimento prioriza a eficiência na utilização de recursos computacionais, proporcionando tempos de resposta satisfatórios e uma navegação estável, mesmo em máquinas com configurações mais modestas.

7. REQUISITOS NÃO FUNCIONAIS

A interface acessível e intuitiva tem como objetivo proporcionar uma experiência de utilização simples, eficiente e de fácil compreensão para todos os perfis de usuário. O sistema deve apresentar uma organização clara dos elementos da interface, com menus, botões e funcionalidades distribuídos de forma lógica, permitindo que usuários e bibliotecários realizem suas atividades com facilidade. Além disso, a interface deve seguir princípios de usabilidade e acessibilidade, reduzindo a curva de aprendizado e minimizando a ocorrência de erros durante a utilização do sistema.

A busca inteligente com filtros por título, autor e gênero foi desenvolvida para facilitar a localização das obras cadastradas no acervo. O sistema deve permitir que os usuários realizem pesquisas utilizando diferentes critérios, de forma individual ou combinada, refinando os resultados conforme a necessidade. Além da precisão na recuperação das informações, o mecanismo de busca deve apresentar desempenho satisfatório, fornecendo os resultados em um curto intervalo de tempo, mesmo com o crescimento da base de dados.

O histórico completo de reservas e movimentações do acervo tem como finalidade registrar todas as operações realizadas envolvendo os exemplares da biblioteca. O sistema deve armazenar informações referentes a empréstimos, devoluções, reservas, renovações e demais movimentações efetuadas. Esse registro possibilita maior controle administrativo, facilita a rastreabilidade das operações, auxilia na geração de relatórios gerenciais e contribui para a integridade das informações relacionadas à utilização do acervo.

8. METODOLOGIA DA PESQUISA

A presente pesquisa caracteriza-se como uma pesquisa de campo, cujo objetivo é compreender os processos utilizados por bibliotecários para a organização de livros e catálogos em bibliotecas.

Para isso, foram realizadas entrevistas presenciais com profissionais da área, possibilitando uma análise direta das práticas adotadas no contexto real de trabalho.

Com base em uma pesquisa exploratória previamente realizada, identificou-se a ausência de um software específico que atenda de forma adequada às necessidades de organização e gestão de acervos, o que justifica a relevância e a pertinência deste estudo.

Quanto aos procedimentos técnicos, a pesquisa seguiu o modelo de documentação direta, sendo complementada por pesquisa documental e bibliográfica.

Foram utilizadas bases de dados disponíveis em sites especializados, como os de livrarias, além de arquivos, artigos científicos e demais materiais já existentes que contribuíram para a fundamentação teórica e metodológica da pesquisa.

9. PERGUNTAS E RESPOSTAS

O projeto foi elaborado com o intuito de mapear o fluxo de trabalho, desde a organização do acervo até o controle de usuários. As questões e o diagnóstico resultante estão sintetizados abaixo:

Fluxo Investigado: Quais ferramentas são utilizadas para o controle de entrada e saída (empréstimos)?

Diagnóstico obtido: Uso de planilhas de Excel e registros manuscritos em papel.

Classificação: Como é estruturada a organização física dos livros nas prateleiras?

Diagnóstico obtido: Utilização do "Número de Chamada" (código alfanumérico) na etiqueta da lombada.

Expectativa Técnica: De que forma uma aplicação dedicada poderia otimizar a rotina da biblioteca?

Diagnóstico obtido: Por meio da centralização de dados e maior controle sobre a disponibilidade do acervo.

Requisitos Desejados: Quais funções são consideradas essenciais para o novo sistema?

Diagnóstico obtido: Busca avançada (título/autor), monitoramento de prazos e alertas automáticos de atraso.

Pontos Críticos: Quais as maiores dificuldades enfrentadas na gestão atual?

Diagnóstico obtido: Alta incidência de erros por registros manuais e dificuldade em rastrear livros não devolvidos.

10. TRABALHOS RELACIONADOS

A fim de embasar o desenvolvimento do presente projeto, foram analisados sistemas digitais de bibliotecas já existentes que abordam problemáticas semelhantes ao gerenciamento de acervos bibliográficos.

Os sistemas abaixo servem como referência e ponto de comparação para a proposta desenvolvida neste trabalho.

BIBLION - Biblioteca Digital do Estado de São Paulo, disponível em <<https://biblion.org.br/>>: Trata-se de uma iniciativa do Governo do Estado de São Paulo, por meio da Secretaria da Cultura, Economia e Indústria Criativas, sob a gestão da SP Leituras.

A plataforma oferece acesso gratuito a um acervo com mais de 20 mil títulos digitais, incluindo livros, audiolivros, podcasts e programação cultural, disponível a qualquer pessoa residente no território nacional.

Entre seus recursos destacam-se a busca avançada por título, autor e tema, o empréstimo de até duas obras simultâneas por 15 dias, além de clubes de leitura e atividades de formação.

A BIBLION representa um modelo consolidado de democratização do acesso à leitura por meios digitais, sendo uma referência relevante para o presente projeto, especialmente no que diz respeito à acessibilidade e organização do acervo.

BIBLIOSP Digital – Prefeitura de São Paulo, disponível em <<https://bibliotecacircula.prefeitura.sp.gov.br/pesquisa/>>: Foi desenvolvida pela Coordenação do Sistema Municipal de Bibliotecas (CSMB) da Secretaria Municipal de Cultura da cidade de São Paulo.

Trata-se de uma plataforma que reúne o acervo das 54 bibliotecas públicas municipais e dos 29 pontos e bosques de leitura da rede municipal.

Por meio dela, é possível realizar buscas avançadas por título, autor, gênero literário, formato e idioma, além de verificar a disponibilidade e a quantidade de exemplares em cada unidade.

A plataforma oferece ainda acesso a livros digitais gratuitos por meio do aplicativo Árvore.

Sua estrutura de catalogação e controle de acervo distribuído serve como inspiração para a proposta de gestão centralizada do presente projeto.

<Biblioteca Virtual Pearson, disponível em <https://www.bvirtual.com.br/>>: É uma plataforma ativa desde 2005 e está voltada principalmente às instituições de ensino superior, oferecendo um acervo de mais de 17 mil e-books técnicos e científicos em língua portuguesa e inglesa, de diversas editoras parceiras.

A plataforma conta com recursos como busca inteligente com filtros, metas de leitura, listas personalizadas, leitura offline e sugestões baseadas em inteligência artificial.

Embora seja um serviço pago e direcionado ao ensino superior, a Biblioteca Virtual Pearson demonstra o potencial das plataformas digitais para ampliar o acesso ao conhecimento de forma organizada e eficiente, servindo como referência em termos de usabilidade e funcionalidades para o sistema proposto neste trabalho.

A análise desses sistemas reforça a relevância e a pertinência do presente projeto, evidenciando que, embora existam soluções digitais para grandes redes de bibliotecas, ainda há uma lacuna significativa no que diz respeito a ferramentas acessíveis, simples e adequadas à realidade das bibliotecas escolares públicas de pequeno porte, como as das ETEC's.

11. METODOLOGIA DO TRABALHO

O desenvolvimento do projeto seguiu uma abordagem baseada na metodologia ágil, utilizando como referência os princípios do Scrum adaptado ao contexto escolar.

As atividades foram organizadas em ciclos curtos de desenvolvimento (sprints), permitindo entregas incrementais e revisões constantes do produto.

O planejamento e o acompanhamento das tarefas foram realizados por meio da ferramenta Trello, que possibilitou a visualização do fluxo de trabalho no modelo Kanban, com colunas representando os estados: A Fazer, Em Andamento e Concluído.

Essa abordagem favoreceu a distribuição equitativa das responsabilidades entre os integrantes da equipe e facilitou a identificação de impedimentos durante o processo.

As reuniões de alinhamento foram realizadas periodicamente, tanto de forma presencial quanto remota, garantindo a comunicação eficiente entre os membros do grupo.

A cada sprint, foram definidas metas específicas relacionadas ao levantamento de requisitos, ao design das telas, ao desenvolvimento do back-end e do front-end, e à documentação do sistema.

A prototipação das interfaces foi conduzida no Figma, permitindo validação visual antes da implementação. O controle de versão do código foi gerenciado com Git e GitHub, assegurando o histórico de alterações e a colaboração segura entre os desenvolvedores.

12. FERRAMENTAS E LINGUAGENS UTILIZADAS

Linguagens de Back-end

- **SQL**: Linguagem utilizada para consultas e manipulação dos dados armazenados no banco de dados relacional.
- **JavaScript (JS)**: Linguagem de programação utilizada no servidor por meio do ambiente de execução Node.js, responsável pela lógica de negócio, processamento das requisições HTTP e comunicação com o banco de dados MySQL.

Linguagens de Front-end

- **HTML**: Linguagem de marcação utilizada para estruturar as páginas web do sistema.
- **CSS**: Linguagem de estilização responsável pela aparência visual e responsividade das interfaces.
- **JavaScript (JS)**: Linguagem de programação utilizada para adicionar interatividade e dinamismo às páginas do sistema.

Banco de Dados

- **MySQL**: Sistema Gerenciador de Banco de Dados Relacional (SGBDR) utilizado para armazenar e gerenciar todas as informações do sistema, como cadastro de livros, usuários e movimentações de empréstimo.

Versionamento

- **Git**: Ferramenta de controle de versão utilizada para registrar o histórico de alterações no código-fonte.
- **GitHub**: Plataforma de hospedagem de repositórios Git, utilizada para colaboração entre os membros da equipe e armazenamento remoto do projeto.

Ferramentas

- **Visual Studio Code (VS Code)**: Editor de código utilizado pela equipe no desenvolvimento do sistema.
- **Figma**: Ferramenta de design utilizada para a criação dos protótipos e wireframes das interfaces do sistema.
- **Trello**: Plataforma de gerenciamento de tarefas utilizada para organizar o fluxo de trabalho da equipe com base na metodologia Kanban.

13. CRONOGRAMA DO TRABALHO

Atividades	Meses										
	Agosto	Setembro	Outubro	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio	Junho
Criação de ideias do Projeto											
Levantamento dos Requisitos											
Definição do Tema											
Pesquisa de Campo											
Pesquisa Bibliográfica											
Fluxograma											
Design do Protótipo											
Escolha das Linguagens											
Diagrama de Caso de uso											
Banco de Dados (Esboço)											
Documentação											
Programação do Projeto											
Testes e Finalização											
Apresentação do TCC											

Figura 1 - Cronograma de atividades do projeto, distribuído por bimestre, com as etapas de levantamento de requisitos, desenvolvimento, testes e documentação.

14. DOCUMENTAÇÃO DO SOFTWARE

14.1. FLUXOGRAMA

Um fluxograma é um tipo de diagrama que explica visualmente um processo ou fluxo de trabalho. Segue abaixo o fluxograma desenhado conforme pesquisas de campo que foram realizadas.

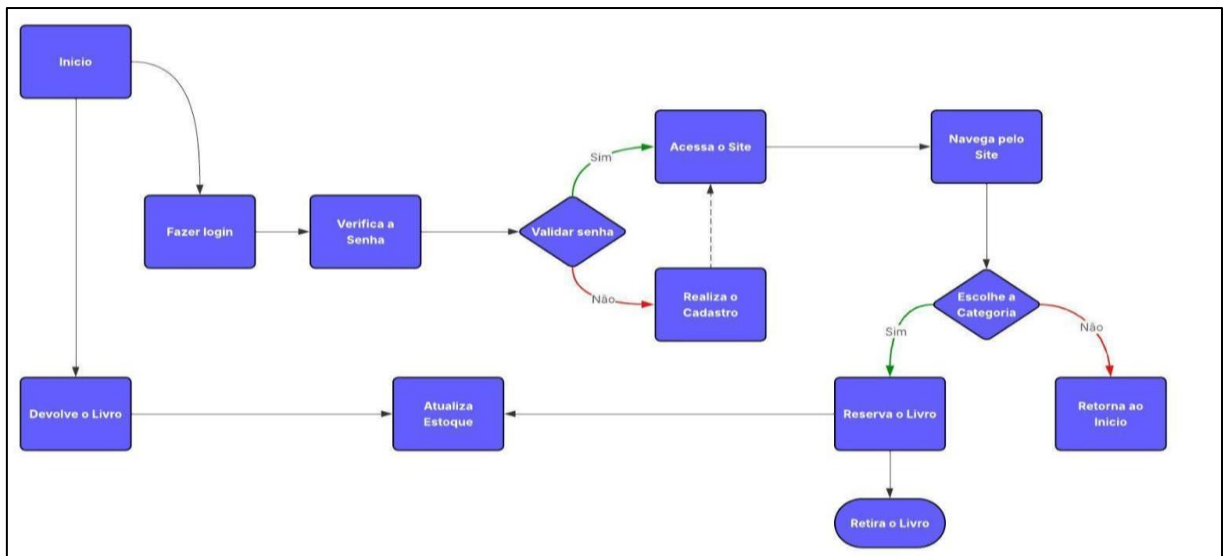


Figura 2 - Fluxograma do sistema de gestão de biblioteca, representando o fluxo de ações do usuário e do administrador, desde o acesso ao catálogo até a conclusão do empréstimo e devolução.

15. DIAGRAMA DE CASO DE USO

Na Linguagem de Modelagem Unificada (UML), o diagrama de caso de uso resume os detalhes dos usuários do sistema (também conhecidos como atores) e as interações deles com o sistema.

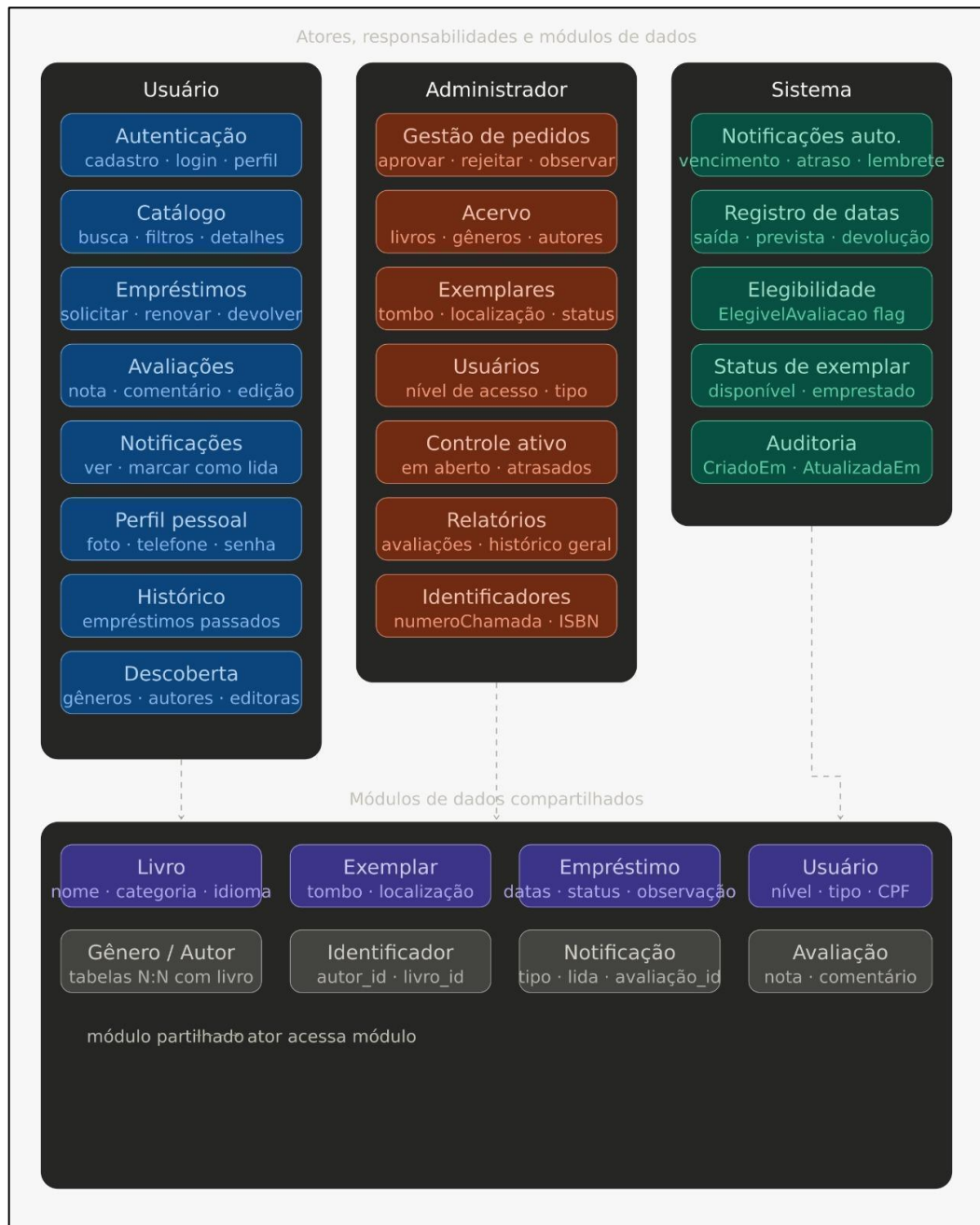


Figura 3 - Diagrama de caso de uso do sistema, ilustrando os atores (Usuário Leitor e Administrador) e as funcionalidades disponíveis para cada perfil, como solicitar empréstimo, devolver livro, gerenciar acervo e aprovar solicitações.

16. MODELO CONCEITUAL DO BANCO DE DADOS

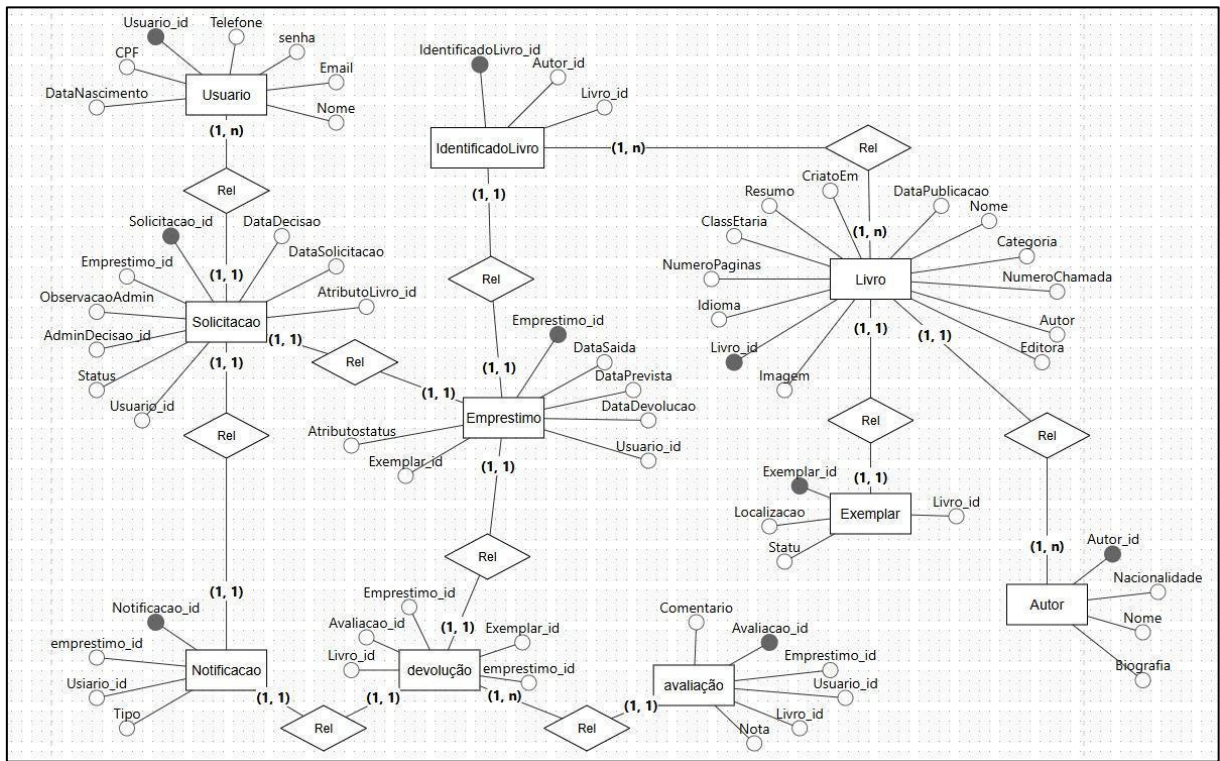


Figura 4 - Modelo Conceitual do banco de dados, representando as entidades principais do sistema e seus relacionamentos em alto nível, sem especificação de tipos de dados ou chaves.

O modelo conceitual tem como objetivo representar, de forma clara e acessível, as principais entidades do sistema e como elas se relacionam entre si, sem ainda se preocupar com detalhes técnicos de implementação.

No contexto deste projeto, as entidades centrais são Usuário, Livro, Exemplar, Empréstimo, Solicitação de Empréstimo e Notificação.

De forma simplificada, um usuário pode solicitar empréstimos de livros, cada livro pode ter vários exemplares físicos com disponibilidades independentes, e cada empréstimo é vinculado a um único usuário e a um único exemplar.

Esse conjunto de relações garantiu que as regras do sistema, como impedir o empréstimo de um exemplar já ocupado, fossem refletidas diretamente na estrutura do banco de dados.

17. MODELO LÓGICO DO BANCO DE DADOS

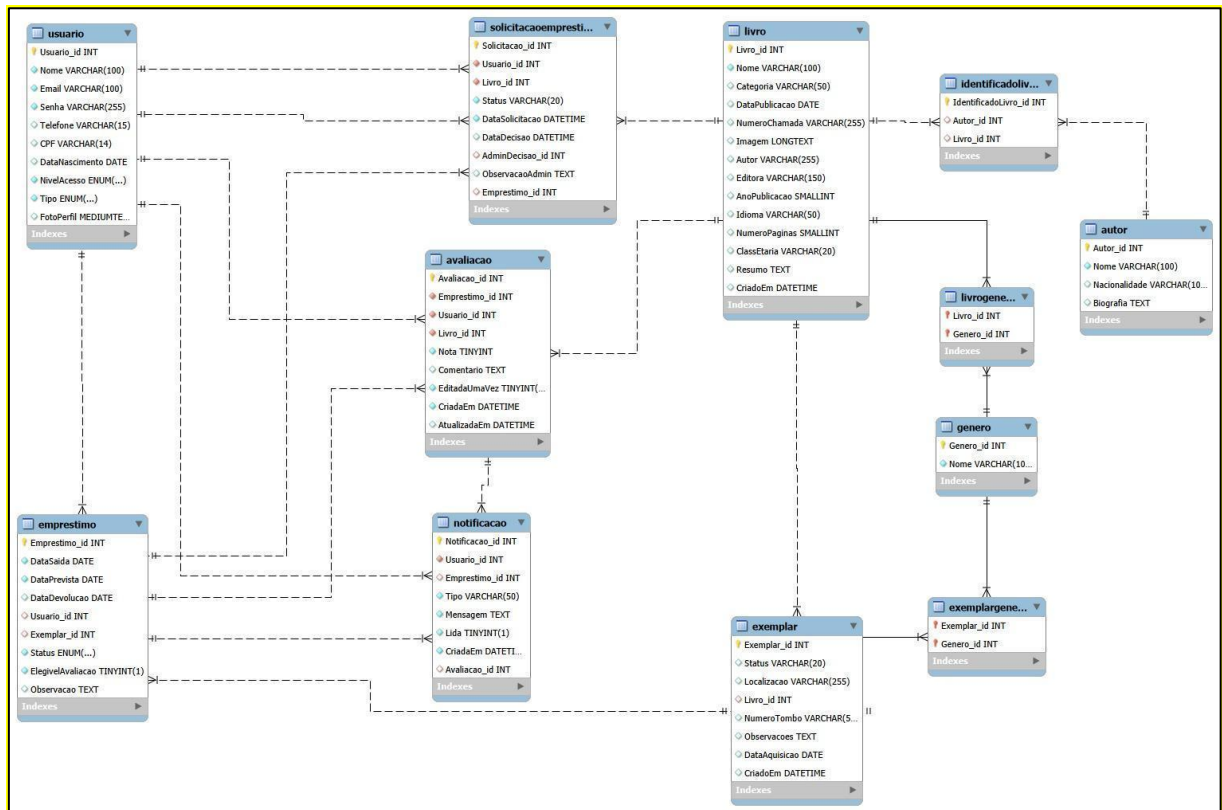


Figura 5 - Modelo Lógico do banco de dados, apresentando as tabelas, atributos, chaves primárias e estrangeiras que estruturam o armazenamento das informações do sistema.

O modelo lógico traduz as entidades conceituais em tabelas relacionais, definindo colunas, tipos de dados e os vínculos entre elas por meio de chaves primárias e estrangeiras.

A tabela Usuário armazena os dados de cadastro e distingue leitores de administradores pelo campo Tipo.

A tabela Livro guarda os metadados bibliográficos, enquanto a tabela Exemplar representa cada cópia física disponível no acervo.

A tabela Empréstimo registra as movimentações de saída e retorno, com datas e status (ativo, devolvido ou atrasado).

Já a tabela Solicitação Empréstimo controla o fluxo de aprovação dos pedidos, e a tabela Notificação armazena as mensagens geradas automaticamente pelo sistema para cada usuário.

18. ARQUITETURA DO SISTEMA

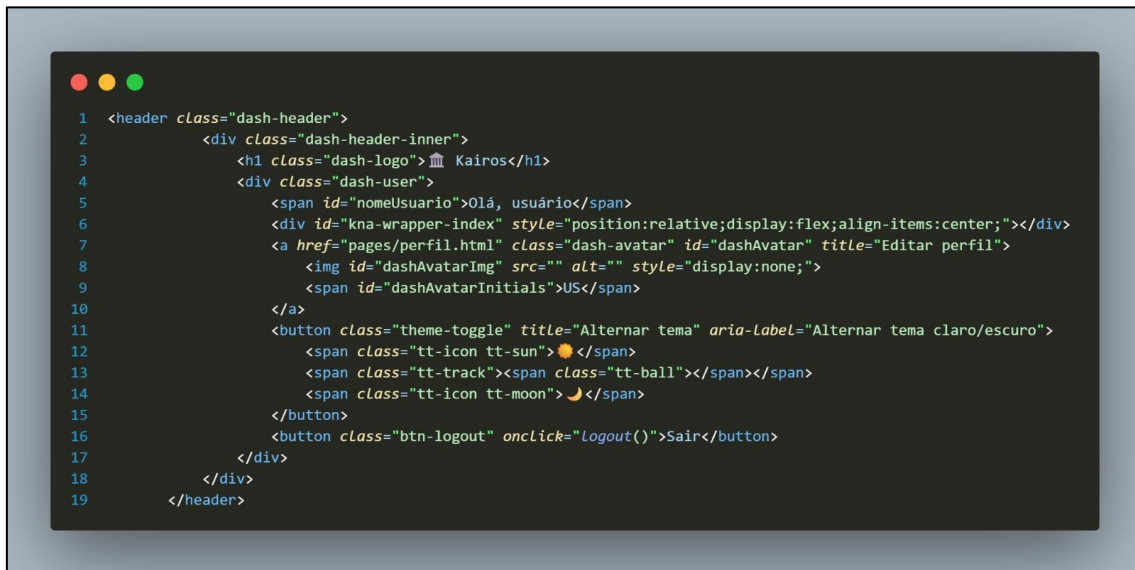


Figura 6 - Diagrama de arquitetura geral do sistema, evidenciando a separação entre front-end, back-end e banco de dados e a comunicação entre as camadas via requisições HTTP.

O sistema foi construído sobre o padrão cliente-servidor, com três camadas bem definidas: front-end, back-end e banco de dados.

O front-end é formado por páginas HTML com CSS e JavaScript puro, sem frameworks. Ele se comunica com o servidor por meio de requisições HTTP assíncronas, trafegando dados em JSON.

O back-end, desenvolvido em Node.js com Express, expõe uma API REST organizada por recurso e aplica todas as regras de negócio antes de consultar ou atualizar o banco.

O MySQL armazena todas as informações persistentes, escolhido pela necessidade de integridade referencial entre os dados.

Essa separação em camadas traz uma vantagem prática importante: alterações em uma parte do sistema não afetam as demais, facilitando a manutenção e evolução do projeto.

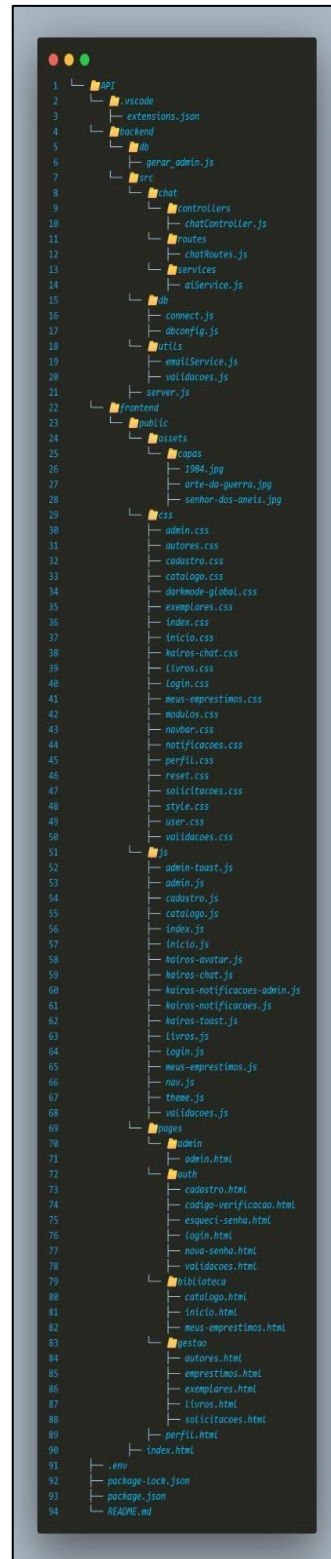


Figura 7 - Estrutura de pastas do projeto, com a organização dos diretórios de front-end e back-end.

```
1 <header class="dash-header">
2   <div class="dash-header-inner">
3     <h1 class="dash-logo"> 🏰 Kairos</h1>
4     <div class="dash-user">
5       <span id="nomeUsuario">Olá, usuário</span>
6       <div id="kna-wrapper-index" style="position:relative;display:flex;align-items:center;"></div>
7       <a href="pages/perfil.html" class="dash-avatar" id="dashAvatar" title="Editar perfil">
8         <img id="dashAvatarImg" src="" alt="" style="display:none;">
9         <span id="dashAvatarInitials">US</span>
10      </a>
11      <button class="theme-toggle" title="Alternar tema" aria-label="Alternar tema claro/escuro">
12        <span class="tt-icon tt-sun"> ☀️</span>
13        <span class="tt-track"><span class="tt-ball"></span></span></span>
14        <span class="tt-icon tt-moon"> 🌙</span>
15      </button>
16      <button class="btn-logout" onClick="Logout()">Sair</button>
17    </div>
18  </div>
19 </header>
```

Figura 8 - Diagrama de fluxo de requisição e resposta entre o cliente e o servidor Node.js, passando pelo roteamento Express e a camada de acesso ao banco MySQL.

```

const PRAZO_EMPRESTIMO_DIAS = 14;

// Calcula a DataPrevista somando 14 dias a partir da data de saída
function calcularDataPrevista(dataBase = new Date()) {
  const d = new Date(dataBase);
  d.setDate(d.getDate() + PRAZO_EMPRESTIMO_DIAS);
  return d.toISOString().split('T')[0];
}

// Diferença em dias entre duas datas 'YYYY-MM-DD' (sem bug de fuso horário)
function diffDiasDatas(dataInicioStr, dataFimStr) {
  const [iy, im, id] = dataInicioStr.split('-').map(Number);
  const [fy, fm, fd] = dataFimStr.split('-').map(Number);
  return Math.round((Date.UTC(fy, fm - 1, fd) - Date.UTC(iy, im - 1, id)) / 86400000);
}

// Mensagem de vencimento próximo (deixa claro que o prazo é de 14 dias)
function montarMensagemVencimentoProximo(nomeLivro, dataPrevista, diasRestantes) {
  const diasTexto = diasRestantes <= 0 ? 'hoje' : `em ${diasRestantes} dia(s)`;
  return `Seu empréstimo do livro "${nomeLivro}" vence ${diasTexto} (${dataPrevista}). ` +
    `Lembrete: o prazo de empréstimo é de ${PRAZO_EMPRESTIMO_DIAS} dias a partir da retirada. ` +
    `Por favor, devolva o livro até a data prevista.`;
}

// Mensagem de atraso (deixa claro que o prazo era de 14 dias e a devolução está pendente)
function montarMensagemAtraso(nomeLivro, dataPrevista, diasAtraso) {
  const diasTexto = diasAtraso === 1 ? '1 dia' : `${diasAtraso} dias`;
  return `O prazo de devolução do livro "${nomeLivro}" está vencido. ` +
    `O prazo de empréstimo era de ${PRAZO_EMPRESTIMO_DIAS} dias, com devolução prevista para ${dataPre
    `e a devolução está pendente há ${diasTexto}. Por favor, devolva o livro o quanto antes para evita

```

Figura 9 - O Código que valida a mensagem de atraso e o prazo de 14 dias para a devolução.

19. TELAS DO SISTEMA

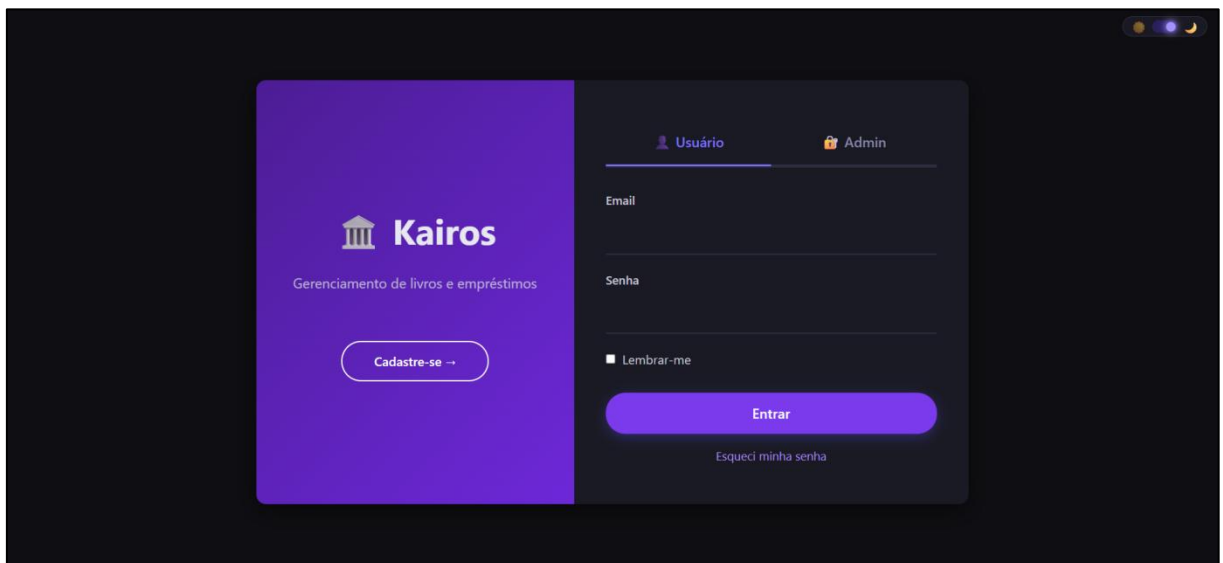


Figura 10 - Tela de login do sistema, com campos de e-mail e senha e acesso diferenciado para usuários comuns e administradores.

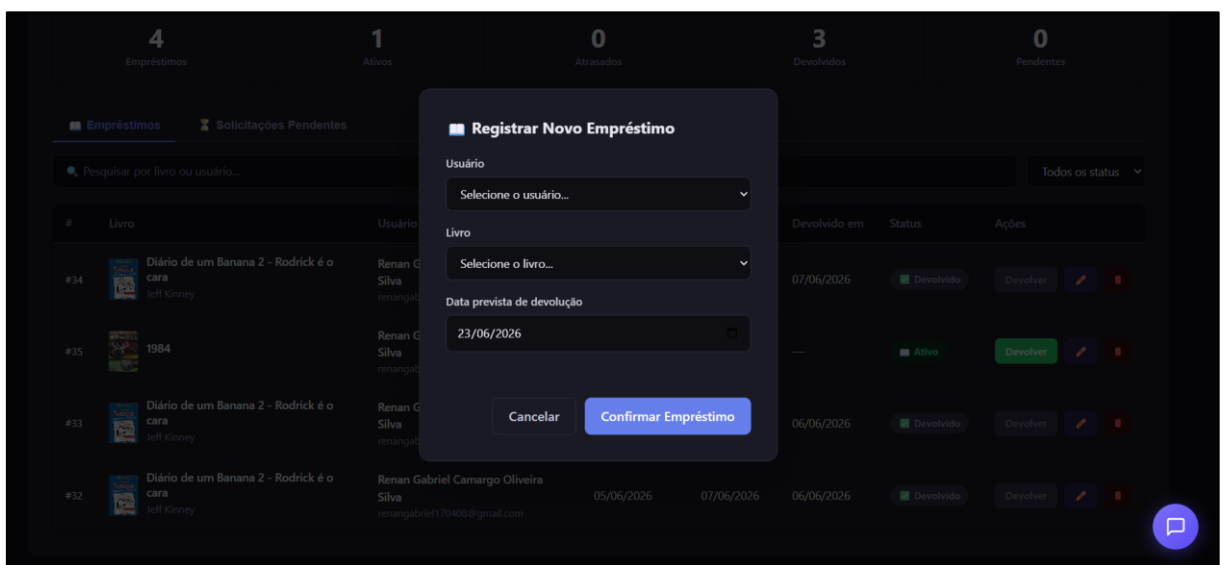


Figura 11 - Tela de catálogo de livros, exibindo o acervo disponível com opção de busca e botão para solicitação de empréstimo.

Tela de Login: Ponto de entrada do sistema, onde usuários e administradores informam e-mail e senha para autenticação. Após a validação, cada perfil é direcionado para sua área correspondente.

Tela de Catálogo: Apresenta o acervo completo com busca por título ou autor. Ao selecionar um livro, o usuário visualiza seus detalhes e, se houver exemplar disponível, pode enviar uma solicitação de empréstimo.

Tela de Solicitações (Administrador): Lista os pedidos de empréstimo pendentes. O administrador pode aprovar ou reprovar cada solicitação; ao aprovar, o empréstimo é criado automaticamente e o usuário é notificado.

Tela de Empréstimos (Administrador): Exibe os empréstimos ativos e atrasados. O administrador registra a devolução dos livros, o que libera o exemplar no acervo e gera uma notificação ao leitor.

Controle completo de empréstimos e devoluções

+ Novo Empréstimo

4 Empréstimos 1 Ativos 0 Atrasados 3 Devolvidos 0 Pendentes

Empréstimos Solicitações Pendentes

Pesquisar por livro ou usuário... Todos os status

#	Livro	Usuário	Emprestado em	Devolver até	Devolvido em	Status	Ações
#34	Diário de um Banana 2 - Rodrick é o cara Jeff Kinney	Renan Gabriel Camargo Oliveira Silva renangabriel170408@gmail.com	07/06/2026	08/06/2026	07/06/2026	Devolvido	Devolver ✎ 🚫
#35	1984	Renan Gabriel Camargo Oliveira Silva renangabriel170408@gmail.com	07/06/2026	21/06/2026	—	Ativo	Devolver ✎ 🚫
#33	Diário de um Banana 2 - Rodrick é o cara Jeff Kinney	Renan Gabriel Camargo Oliveira Silva renangabriel170408@gmail.com	06/06/2026	20/06/2026	06/06/2026	Devolvido	Devolver ✎ 🚫
#32	Diário de um Banana 2 - Rodrick é o cara Jeff Kinney	Renan Gabriel Camargo Oliveira Silva renangabriel170408@gmail.com	05/06/2026	07/06/2026	06/06/2026	Devolvido	Devolver ✎ 🚫

Figura 12 - Tela de gerenciamento de solicitações de empréstimo, acessível ao administrador, com opções de aprovação ou reprovação de cada pedido.

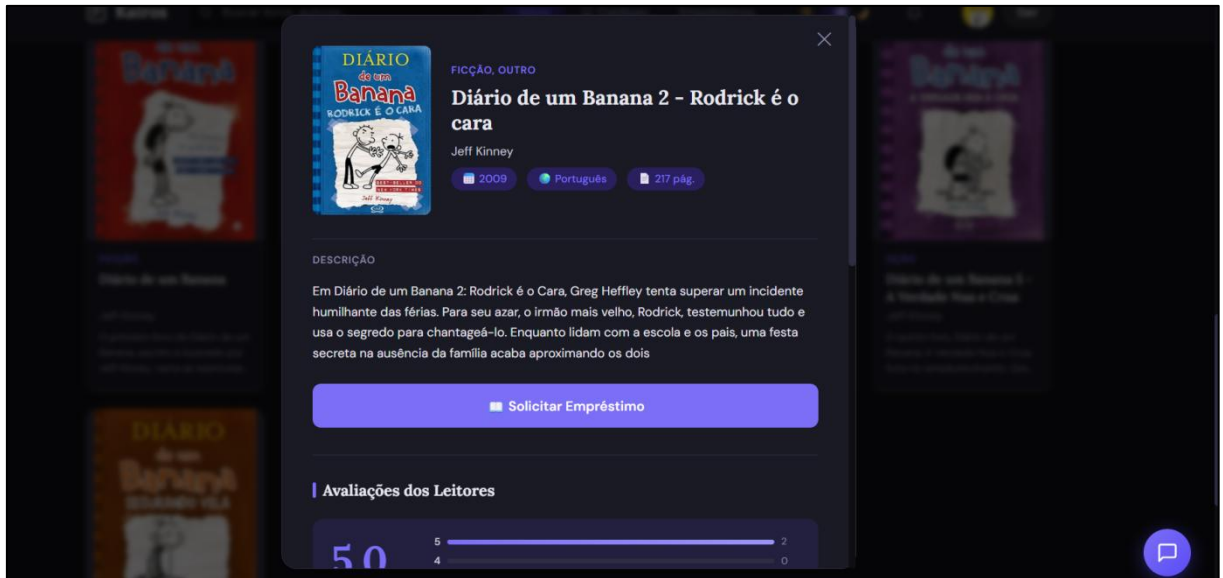


Figura 13 - Tela de gerenciamento de empréstimos ativos, com informações de prazo, status e botão para registro de devolução.

20. CÓDIGOS-FONTE DO SISTEMA

```
//
// LOGIN - ☒ COMPARAÇÃO COM HASH (bcrypt.compare)
//
app.post('/usuario/login', (req, res) => {
  const { Email, Senha } = req.body;
  if (!Email || !Senha) {
    return res.status(400).json({ error: 'Email e senha são obrigatórios.' });
  }

  dbconfig.query(
    'SELECT * FROM Usuario WHERE Email = ?',
    [Email],
    async (err, results) => {
      if (err) return handleQuery(res, err);
      if (results.length === 0) {
        return res.status(401).json({ error: 'Credenciais inválidas.' });
      }

      const usuario = results[0];
      // ☒ CORREÇÃO 3: comparar senha com hash
      const senhaCorreta = await bcrypt.compare(Senha, usuario.Senha);
      if (!senhaCorreta) {
        return res.status(401).json({ error: 'Credenciais inválidas.' });
      }

      const { Senha: _, ...usuarioSemSenha } = usuario;
      res.json({ usuario: usuarioSemSenha });
    }
  );
});
```

Figura 14 - Código-fonte do back-end: rota de login de usuário com validação de credenciais e comparação de senha via bcrypt.

Autenticação (Login): As rotas de login recebem e-mail e senha, verificam as credenciais no banco com `bcrypt.compare` e retornam os dados do usuário sem a senha. No front-end, a resposta é usada para redirecionar o usuário a área correta conforme seu perfil.

Solicitação de Empréstimo: Antes de registrar um pedido, o sistema verifica se já existe solicitação pendente, empréstimo ativo e exemplar disponível. Somente após essas verificações a solicitação é criada e os administradores são notificados.

Aprovação e Criação do Empréstimo: Ao aprovar, o sistema localiza um exemplar disponível, cria o empréstimo com prazo de 14 dias, atualiza o status do exemplar e notifica o usuário com a data de devolução prevista.

Devolução: O administrador confirma a devolução, o sistema atualiza o empréstimo, libera o exemplar no acervo e envia uma notificação ao usuário convidando-o a avaliar o livro.

```
//
// ADMIN LOGIN - ☒ COMPARAÇÃO COM HASH (bcrypt.compare)
//
app.post('/admin/login', (req, res) => {
  const { Email, Senha } = req.body;
  if (!Email || !Senha) {
    return res.status(400).json({ error: 'Email e senha são obrigatórios.' });
  }

  dbconfig.query(
    'SELECT * FROM Usuario WHERE Email = ? AND Tipo = "admin"',
    [Email],
    async (err, results) => {
      if (err) return handleQuery(res, err);
      if (results.length === 0) {
        return res.status(401).json({ error: 'Credenciais inválidas ou sem permissão de administrado' });
      }

      const admin = results[0];
      // ☒ CORREÇÃO 4: comparar senha com hash
      const senhaCorreta = await bcrypt.compare(Senha, admin.Senha);
      if (!senhaCorreta) {
        return res.status(401).json({ error: 'Credenciais inválidas ou sem permissão de administrado' });
      }

      const { Senha: _, ...adminSemSenha } = admin;
      res.json({ usuario: adminSemSenha });
    }
  );
});
```

Figura 15 - Código-fonte do back-end: rota de solicitação de empréstimo, com verificações de duplicidade, disponibilidade de exemplar e notificação aos administradores.

```

<!-- Aba Usuário -->
<div id="usuario" class="tab-content active">
  <div id="usuarioAlert"></div>
  <form id="usuarioForm">
    <div class="form-group">
      <label for="usuarioEmail">Email</label>
      <input type="email" id="usuarioEmail" required>
    </div>
    <div class="form-group">
      <label for="usuarioSenha">Senha</label>
      <input type="password" id="usuarioSenha" required>
    </div>
    <div class="remember-me">
      <input type="checkbox" id="usuarioLembrar">
      <label for="usuarioLembrar" style="margin:0;">Lembrar-me</label>
    </div>
    <button type="submit" class="btn-submit" id="btnUsuario">Entrar</button>
  </form>
  <div class="forgot-link"><a href="esqueci-senha.html">Esqueci minha senha</a></div>
</div>

```

Figura 16 - Código-fonte do back-end: rota de aprovação de solicitação, com criação do empréstimo, atualização do exemplar e notificação ao usuário.

```

<div class="right-side">
  <!-- Aba Admin -->
  <div id="admin" class="tab-content">
    <div id="adminAlert"></div>
    <form id="adminForm">
      <div class="form-group">
        <label for="adminEmail">Email Admin</label>
        <input type="email" id="adminEmail" required>
      </div>
      <div class="form-group">
        <label for="adminSenha">Senha</label>
        <input type="password" id="adminSenha" required>
      </div>
      <button type="submit" class="btn-submit" id="btnAdmin">Entrar como Admin</button>
    </form>
  </div>
</div>
</div>
<script>
  const API = 'http://localhost:3000';

  (function() {
    const token = sessionStorage.getItem('authToken');
    const tipo = sessionStorage.getItem('tipoUsuario');
    if (token && tipo) {
      // Admin vai direto para o painel; usuário comum para validação de conta
      if (tipo === 'admin') {
        window.location.href = '../index.html';
      } else {
        window.location.href = 'validacoes.html';
      }
    }
  })

```

Figura 17 - Código-fonte do back-end: rota de devolução de livro, com atualização do status do empréstimo, liberação do exemplar e geração de notificação de avaliação.

```
// — Solicitar empréstimo —
async function solicitarEmprestimo(livroId, titulo) {
  const usuarioId = sessionStorage.getItem('usuarioId');
  if (!usuarioId) {
    mostrarToast('Faça login para solicitar um empréstimo.', 'error');
    return;
  }

  const btn = document.getElementById('btnSolicitar');
  if (btn) { btn.disabled = true; btn.textContent = 'Enviando solicitação...'; }

  try {
    const res = await fetch(`${API}/solicitacao`, {
      method : 'POST',
      headers: { 'Content-Type': 'application/json' },
      body    : JSON.stringify({ Usuario_id: parseInt(usuarioId), Livro_id: livroId })
    });
    const data = await res.json();
    fecharModal();
    if (res.ok) {
      mostrarToast(`📄 Solicitação de "${titulo}" enviada! Aguarde a aprovação.`, 'success');
    } else {
      mostrarToast(data.error || 'Não foi possível enviar a solicitação.', 'error');
    }
  } catch {
    fecharModal();
    mostrarToast('Erro de conexão com o servidor.', 'error');
  }
}
```

Figura 18 - Código-fonte do front-end: funções JavaScript responsáveis pelo envio das requisições de login, solicitação de empréstimo e registro de devolução.

```

const PRAZO_EMPRESTIMO_DIAS = 14;

// Calcula a DataPrevista somando 14 dias a partir da data de saída
function calcularDataPrevista(dataBase = new Date()) {
  const d = new Date(dataBase);
  d.setDate(d.getDate() + PRAZO_EMPRESTIMO_DIAS);
  return d.toISOString().split('T')[0];
}

// Diferença em dias entre duas datas 'YYYY-MM-DD' (sem bug de fuso horário)
function diffDiasDatas(dataInicioStr, dataFimStr) {
  const [iy, im, id] = dataInicioStr.split('-').map(Number);
  const [fy, fm, fd] = dataFimStr.split('-').map(Number);
  return Math.round((Date.UTC(fy, fm - 1, fd) - Date.UTC(iy, im - 1, id)) / 86400000);
}

// Mensagem de vencimento próximo (deixa claro que o prazo é de 14 dias)
function montarMensagemVencimentoProximo(nomeLivro, dataPrevista, diasRestantes) {
  const diasTexto = diasRestantes <= 0 ? 'hoje' : `em ${diasRestantes} dia(s)`;
  return `Seu empréstimo do livro "${nomeLivro}" vence ${diasTexto} (${dataPrevista}). ` +
    `Lembrete: o prazo de empréstimo é de ${PRAZO_EMPRESTIMO_DIAS} dias a partir da retirada. ` +
    `Por favor, devolva o livro até a data prevista.`;
}

// Mensagem de atraso (deixa claro que o prazo era de 14 dias e a devolução está pendente)
function montarMensagemAtraso(nomeLivro, dataPrevista, diasAtraso) {
  const diasTexto = diasAtraso === 1 ? '1 dia' : `${diasAtraso} dias`;
  return `O prazo de devolução do livro "${nomeLivro}" está vencido. ` +
    `O prazo de empréstimo era de ${PRAZO_EMPRESTIMO_DIAS} dias, com devolução prevista para ${dataPre
    `e a devolução está pendente há ${diasTexto}. Por favor, devolva o livro o quanto antes para evita

```

Figura 19 - Código que representa a regra dos 14 dias a partir do atraso na devolução do livro.

21. CICLO DE VIDA DO PROGRAMA

21.1. SOLICITAÇÕES DE EMPRÉSTIMO

Usuário solicita empréstimo e o usuário *admin* aprova ou reprova. Ao aprovar, um empréstimo é criado automaticamente com exemplar disponível e então é gerada uma notificação para o leitor.

21.2. NOTIFICAÇÕES

Os usuários do sistema, que podem ser *admin* ou *leitor*, conseguem marcar notificação por notificação como lida ou não lida. Ou se desejarem, podem marcar todas de uma vez.

21.3. CADASTRO COM VALIDAÇÃO DE CPF, EMAIL E SENHA – HASH COM BSCRYPT

Autenticação por sessão de navegador para usuários comuns e administradores, com senhas armazenadas com hash bcrypt.

A recuperação de senha segue três etapas: o usuário solicita um código enviado por email com validade de 10 minutos; após verificar o código, o backend autoriza a redefinição por 15 minutos; a nova senha deve ter no mínimo 8 caracteres, com letras maiúsculas, minúsculas e números. Por segurança, o sistema não confirma se o email está cadastrado ao receber a solicitação.

A atualização de perfil permite alterar nome, foto e senha. Email, CPF, telefone e data de nascimento são imutáveis por este endpoint.

22. CONSIDERAÇÕES FINAIS

O trabalho apresentou o desenvolvimento do sistema Biblioteca Online, criado para automatizar a gestão de bibliotecas e substituir métodos manuais de controle.

Utilizando a ETEC Jardim Ângela como referência, o sistema foi desenvolvido com arquitetura MVC, Node.js e MySQL, permitindo sua adaptação a diferentes tipos de bibliotecas.

Os objetivos propostos foram alcançados por meio da implementação de funcionalidades como cadastro de livros, controle de empréstimos, devoluções, autenticação de usuários e notificações.

Além disso, o projeto contribuiu para a formação técnica dos autores, possibilitando a aplicação prática dos conhecimentos adquiridos no curso.

Como melhorias futuras, sugerem-se notificações por e-mail, relatórios gerenciais, personalização da identidade visual e uma versão mobile.

Conclui-se que a Biblioteca Online é uma solução acessível e eficiente para modernizar a gestão de bibliotecas.

23. REFERÊNCIAS

- BIBLION. **BibliON — Biblioteca Digital Gratuita do Estado de São Paulo**. Secretaria da Cultura, Economia e Indústria Criativas do Estado de São Paulo; SP Leituras. Disponível em: <https://biblion.org.br/>. Acesso em: 2025.
- PREFEITURA DE SÃO PAULO. **BiblioSP Digital — Catálogo Eletrônico do Sistema Municipal de Bibliotecas**. Secretaria Municipal de Cultura e Economia Criativa. Disponível em: <https://bibliotecacircula.prefeitura.sp.gov.br/pesquisa/>. Acesso em: 2025.
- PEARSON. **Biblioteca Virtual Pearson**. Disponível em: <https://www.bvirtual.com.br/>. Acesso em: 2025.
- ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.
- PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: McGraw-Hill, 2016.
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson Education do Brasil, 2018.
- FLANAGAN, David. **JavaScript: o guia definitivo**. 6. ed. Porto Alegre: Bookman, 2013.
- TILKOV, Stefan; VINOSKI, Steve. Node.js: Using JavaScript to Build High-Performance Network Programs. **IEEE Internet Computing**, v. 14, n. 6, p. 80-83, nov./dez. 2010.
- FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas**. Porto Alegre: Bookman, 2006.
- FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. Tese (Doutorado) — University of California, Irvine, 2000. Disponível em:

<https://ics.uci.edu/~fielding/pubs/dissertation/top.htm>. Acesso em: 2025.

- SCHWABER, Ken; SUTHERLAND, Jeff. **O Guia do Scrum: as regras do jogo**. Scrum.org, 2020. Disponível em: <https://scrumguides.org>. Acesso em: 2025.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. 8. ed. Rio de Janeiro: Elsevier, 2004.