



**ESCOLA SENAI “A. JACOB LAFER”
TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**BRUNA DOS SANTOS ALMEIDA
GABRIEL SOARES GRECCO
JOÃO MANOEL DA CRUZ FILHO
NATALIA DOS SANTOS ALMEIDA
SARA NOGUEIRA TRANQUILINO DE SOUZA
YASMIN CAPPI SANTOS**

**SISTEMA INTEGRADO DE GESTÃO BIBLIOTECÁRIA PARA APLICATIVO
MÓVEL E PLATAFORMA WEB**

Desenvolvimento para Trabalho de Conclusão de Curso (TCC)

**SANTO ANDRÉ - SP
2025**



**ESCOLA SENAI “A. JACOB LAFER”
TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**BRUNA DOS SANTOS ALMEIDA
GABRIEL SOARES GRECCO
JOÃO MANOEL DA CRUZ FILHO
NATALIA DOS SANTOS ALMEIDA
SARA NOGUEIRA TRANQUILINO DE SOUZA
YASMIN CAPPI SANTOS**

**SISTEMA INTEGRADO DE GESTÃO BIBLIOTECÁRIA PARA APLICATIVO
MÓVEL E PLATAFORMA WEB**

Desenvolvimento para Trabalho de Conclusão de Curso (TCC)

Trabalho de Conclusão de Curso
apresentado à escola SENAI Santo André
"A. Jacob Lafer", como requisito à
obtenção do título de Técnico em
Desenvolvimento de Sistemas.

Área de concentração: Técnico de
Desenvolvimento de Sistemas.

Orientadores: Prof. André Luis Denani e
Prof. Dênis dos Anjos Geres.

**SANTO ANDRÉ - SP
2025**

AGRADECIMENTOS

Gostaríamos de expressar nossa gratidão aos professores, André Luis Denani e Dênis dos Anjos Geres, pela orientação incansável, pelo estímulo intelectual e pelas valiosas contribuições ao longo deste trabalho. Agradecemos também pela paciência ao nos orientar nos momentos de dificuldade, pela disponibilidade em esclarecer dúvidas e pelo incentivo constante que nos motivou a evoluir durante todo o desenvolvimento do projeto.

Durante estes dois longos anos de curso, agradecemos de coração aos colegas de turma e a todos que, direta ou indiretamente, colaboraram para a conclusão deste trabalho.

Agradecemos, também, à instituição SESI-SENAI, pelo suporte financeiro imprescindível no fornecimento de equipamentos de estudo de qualidade, que foram essenciais para a realização deste projeto.

Epígrafe

"A leitura verdadeira me compromete de imediato com o texto que a mim se dá e a que me dou e de cuja compreensão fundamental me vou tornando também sujeito."

(Paulo Freire)

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema web e aplicativo móvel voltado à gestão bibliotecária para alunos e bibliotecários, denominado SiBi. O projeto foi desenvolvido para conclusão do curso de Tecnologia em Análise e Desenvolvimento de Sistemas no curso técnico e integrado do SENAI. O sistema tem como principal objetivo otimizar o gerenciamento do acervo, permitindo o cadastro, consulta e controle de empréstimos e devoluções de livros de forma prática e eficiente. Além disso, o SiBi busca facilitar o acesso dos usuários às informações sobre os exemplares disponíveis, bem como aprimorar a organização interna da biblioteca. Para o desenvolvimento da aplicação, foi realizada uma pesquisa sobre os processos de funcionamento de bibliotecas e os fluxos de sistemas bibliotecários, possibilitando a criação de uma solução digital adaptada às problemáticas apontadas. Atualmente, o sistema encontra-se em sua primeira versão funcional, integrando tanto a plataforma web quanto o aplicativo móvel, voltados à experiência dos usuários e ao apoio das atividades dos bibliotecários.

Palavras-chave: *Sistema web. Aplicativo móvel. Biblioteca. Empréstimos. Acervo.*

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Justificativa	7
1.2	Problemática	8
1.3	Objetivos	8
1.3.1	Objetivo Geral.....	9
1.3.2	Objetivo Específico	9
2	DESENVOLVIMENTO	11
2.1	Biblioteca	11
2.1.1	Biblioteca Virtual	11
2.1.2	Biblioteca Digital.....	12
2.2	Sistemas de Informação Aplicados a Bibliotecas.....	13
2.2.1	Sistema bibliotecário	13
2.2.2	Sistema Web.....	13
2.2.3	Sistema Mobile	14
2.3	Trabalhos Correlatos.....	15
2.3.1	Sistema Biblioteca SophiA	15
2.3.2	Sistema Koha	16
2.3.3	Touch System Tanca	17
2.3.4	Pergamum	18
2.4	Conceitos de Modelagem do Sistema.....	18
2.4.1	Diagrama de caso de uso.....	19
2.4.2	Diagrama entidade-relacionamento (DER)	19
2.5	Ferramentas utilizadas.....	20
2.5.1	Linguagens.....	21
2.5.1.1	HTML	21
2.5.1.2	CSS.....	23
2.5.1.3	JavaScript	25
2.5.1.4	Python	26
2.5.2	GitHub.....	27
2.5.3	Visual Studio Code	28
2.5.4	MySQL Workbench.....	28
2.5.5	Flask.....	28
2.5.6	Flet	29

2.5.7	Ferramentas gráficas	30
2.5.7.1	Canva.....	31
2.5.7.2	Figma	31
2.5.7.3	BoxIcons	31
2.5.7.4	Bootstrap.....	32
2.5.7.5	Draw.io	32
2.6	Ferramentas futuras.....	32
2.6.1	Agente de IA com n8n	33
2.6.2	Framework de desenvolvimento mobile.....	33
2.7	Prototipação	34
2.8	Resultados Obtidos.....	43
2.8.1	Codificação	54
3	METODOLOGIA	62
3.1	Metodologia ágil	62
3.1.1	Metodologia ágil Scrum.....	62
3.2	Levantamento de requisitos	64
3.2.1	Requisitos Funcionais	64
3.2.2	Requisitos Não Funcionais.....	67
3.3	Técnicas de Levantamento de Requisitos	69
3.3.1	Entrevista.....	70
3.3.2	Brainstorming	72
3.4	Priorização de requisitos	73
3.4.1	Tipo de priorização de requisitos	74
3.5	Gráfico de Gantt.....	77
3.6	Modelagem Prática do Sistema	78
3.6.1	Aplicação do diagrama de caso de uso	78
3.6.2	Aplicação do diagrama entidade-relacionamento (DER)	82
3.7	Testes de Software Aplicados ao Sistema	85
CONSIDERAÇÕES FINAIS		87
REFERÊNCIAS.....		88

1 INTRODUÇÃO

A organização e o controle dos livros em bibliotecas escolares são fundamentais para garantir que alunos e professores tenham acesso fácil ao material necessário para o estudo. No entanto, muitas escolas ainda usam métodos manuais, que acabam dificultando o gerenciamento e o empréstimo dos livros.

Este trabalho apresenta o desenvolvimento de um sistema, com aplicativo móvel e plataforma web, para melhorar a gestão da biblioteca escolar. O objetivo é criar uma solução que ajude tanto os alunos a encontrarem os livros quanto os funcionários a controlarem o acervo de forma mais eficiente. O trabalho está dividido em capítulos que explicam a teoria, a criação do sistema, os resultados obtidos e as conclusões finais.

1.1 Justificativa

A opção temática adotada neste Trabalho de Conclusão de Curso (TCC) está fundamentada nas dificuldades observadas no controle de entrada e saída de livros nas bibliotecas escolares, que comprometem o acesso eficiente ao acervo disponível. A carência de livros em determinadas coleções ou unidades, causada pelo não retorno dos exemplares, prejudica diretamente o processo de ensino-aprendizagem e limita o desenvolvimento intelectual dos alunos.

Além disso, embora a instituição disponha de recursos tecnológicos como computadores, tais ferramentas não têm sido plenamente aproveitadas para a otimização da gestão do acervo, o que representa uma oportunidade perdida para modernizar e melhorar os processos internos. Vale destacar que, embora existam plataformas de acervo digital, esses recursos ainda são pouco divulgados e, por isso, pouco acessados pelos alunos. Isso acontece principalmente pela falta de ações efetivas de comunicação e treinamento que possam facilitar o conhecimento e o uso dessas ferramentas digitais. Além disso, a ausência de um software específico para a gestão bibliotecária na própria escola faz com que os processos continuem sendo feitos manualmente, o que resulta em menor eficiência, maior chance de erros e dificuldades no controle dos empréstimos e devoluções de livros.

Portanto, este trabalho se justifica pela necessidade de desenvolver uma solução tecnológica que contribua para a melhoria do atendimento aos alunos e para a otimização das atividades desempenhadas pelos funcionários responsáveis pela gestão bibliotecária. Assim, espera-se não apenas facilitar o acesso aos materiais, mas também promover uma maior valorização do patrimônio bibliográfico e o incentivo à leitura, beneficiando a comunidade escolar e a prática profissional na área de tecnologia da informação aplicada à educação.

1.2 Problemática

A biblioteca é um espaço fundamental para o desenvolvimento acadêmico, social e cultural dentro das instituições de ensino. No entanto, muitas vezes enfrenta desafios relacionados à sua utilização, à percepção de seu papel pela comunidade acadêmica e à eficácia dos serviços oferecidos.

Em muitos casos, ainda se baseiam em processos manuais ou sistemas limitados, o que pode dificultar o gerenciamento das atividades e a interação com os usuários. Essa defasagem tecnológica compromete a eficiência do trabalho dos bibliotecários e a experiência dos leitores, que hoje estão habituados a soluções mais ágeis e acessíveis por meios digitais.

Diante desse cenário, surge a necessidade de desenvolver ferramentas tecnológicas que digitalizem e otimizem as funções da biblioteca, promovendo maior praticidade tanto para os profissionais da área quanto para os usuários. Para entender as reais necessidades e expectativas de ambos os públicos, este trabalho propõe a aplicação de um formulário direcionado a bibliotecários e alunos, com o objetivo de levantar dados sobre as principais dificuldades enfrentadas, o nível de digitalização atual e as funcionalidades desejadas em uma plataforma de gerenciamento e acesso à biblioteca.

1.3 Objetivos

Os objetivos de um projeto acadêmico representam as metas a serem alcançadas na aplicação final, servindo como guia para o desenvolvimento e a avaliação da proposta. A elaboração dos objetivos deste projeto teve como base uma

análise de mercado, avaliando as necessidades da criação de um sistema para gestão no setor bibliotecário SESI. A seguir, serão apresentados o objetivo geral e os objetivos específicos definidos para o trabalho.

1.3.1 Objetivo Geral

Este trabalho busca direcionar o desenvolvimento de um sistema de gestão bibliotecária para escolas SESI que atenda tanto às demandas administrativas quanto às necessidades dos alunos. O objetivo geral deste projeto é elaborar uma solução capaz de modernizar e otimizar as atividades da biblioteca, implementando funcionalidades que permitam melhor gerenciamento do acervo, promovendo autonomia, maior organização e praticidade nos processos internos.

1.3.2 Objetivo Específico

Os objetivos específicos definem as funcionalidades práticas que o sistema deve oferecer, como o gerenciamento completo do acervo, controle de empréstimos e devoluções, reservas e renovações de livros, além de uma interface clara e intuitiva. O sistema utilizará tecnologias atuais para garantir segurança, eficiência e bom funcionamento, contribuindo para a modernização da biblioteca e a melhoria da experiência de seus usuários.

- **Gerenciamento completo do acervo:** Permitir o cadastro, organização e categorização de todas as obras da biblioteca, facilitando a localização e o controle do material disponível.
- **Controle de empréstimos e devoluções:** Automatizar o registro de empréstimos e devoluções, garantindo o acompanhamento correto dos prazos e evitando erros manuais.
- **Reservas e renovações de livros:** Proporcionar aos alunos a possibilidade de reservar obras e renovar empréstimos de forma prática e autônoma, otimizando o uso do acervo.

- **Emissão de relatórios administrativos:** Gerar relatórios sobre movimentação de livros, frequência de empréstimos e obras mais demandadas, auxiliando na tomada de decisões da equipe da biblioteca.
- **Perfis diferenciados de acesso:** Criar contas específicas para bibliotecários e alunos, garantindo que cada usuário tenha acesso apenas às funcionalidades pertinentes ao seu perfil, aumentando a segurança do sistema e tornando a navegação mais intuitiva e eficiente.
 - *Bibliotecários:* Possui acesso total ao sistema, podendo cadastrar, editar e excluir usuários, gerenciar permissões, controlar o acervo completo, registrar e alterar empréstimos e devoluções, emitir relatórios detalhados e realizar ajustes administrativos avançados.
 - *Alunos:* Possui acesso limitado ao sistema, podendo consultar o catálogo online, reservar livros, renovar empréstimos, acessar sua carteirinha digital e visualizar seu histórico de empréstimos.
- **Interface clara e intuitiva:** Desenvolver uma interface amigável e de fácil navegação, permitindo que todos os usuários realizem suas atividades de forma prática e eficiente.
- **Uso de tecnologias atuais:** O sistema será desenvolvido com tecnologias atuais, como Python e Flask no back-end, HTML, CSS, Bootstrap e JavaScript no front-end, além de banco de dados MySQL, assegurando que o sistema funcione de maneira estável, segura e eficiente, contribuindo para a modernização da biblioteca.
- **Garantir segurança de acesso e integridade dos dados:** A segurança será assegurada por autenticação de login, criptografia de senhas e controle de permissões por tipo de usuário (Bibliotecário e Aluno), restringindo o acesso conforme o perfil.
- **Estruturar um ambiente digital com perfis diferenciados (bibliotecários e alunos):** O sistema será projetado para garantir um ambiente seguro e eficaz para todos os usuários, com interfaces adequadas ao perfil de cada um.

2 DESENVOLVIMENTO

Nesta seção, serão apresentados os conceitos essenciais para a compreensão dos temas abordados neste trabalho. Serão discutidos, entre outros, os conceitos de biblioteca virtual, biblioteca digital, automação bibliotecária, e sistemas bibliotecários, além das principais ferramentas utilizadas no contexto da pesquisa. Também será realizada uma análise detalhada dos trabalhos correlatos, visando situar o estudo no panorama atual da área e identificar soluções e abordagens relevantes para o tema.

2.1 Biblioteca

Em um momento da história era raro encontrar registros de informações quando requisitados, entretanto, com o passar do tempo isso se inverteu fazendo com que as informações fossem de fácil acesso, mas muitas vezes não eram organizadas de forma correta. Para facilitar a ordenação e o acesso desses registros ao longo da história os homens os juntaram em um lugar, que com o tempo foi nomeado como biblioteca (MILANESI, 2002).

A biblioteca já foi a forma mais fácil de se obter informações, tratando-se de uma fonte de informação terciária (CUNHA, 2001) contendo diversos livros, que eram emprestados para quem frequentava o local, esse processo por muito tempo foi feito de forma manual até o surgimento da automação.

2.1.1 *Biblioteca Virtual*

As bibliotecas virtuais representam uma importante transformação no acesso à informação e nos hábitos de leitura, sendo caracterizadas por possuírem acervo exclusivamente digital, armazenado em diferentes dispositivos eletrônicos e acessível a partir de qualquer equipamento conectado à internet. Essa característica facilita a gestão, manutenção e atualização constante dos conteúdos disponibilizados.

Diferentemente das bibliotecas físicas, as bibliotecas virtuais não impõem limitações tradicionais como prazos para empréstimo, tempo de espera ou multas por atraso, tornando o acesso à informação mais flexível e democrático. Nesse contexto,

são definidas como uma “biblioteca sem paredes”, expressão que enfatiza a ausência de limitações físicas e reforça a natureza ampliada e acessível desse modelo (ROSA E SILVA, [s.d.] apud PINTO; PORTILHO, 2012). Além da comodidade no acesso, a informação digitalizada pode ser compartilhada instantaneamente e a um custo reduzido, aspecto que configura uma vantagem significativa frente às bibliotecas tradicionais. Tal facilidade promove a democratização da leitura e o incentivo à aprendizagem.

2.1.2 *Biblioteca Digital*

O Sistema de Bibliotecas consiste em uma estrutura organizacional que integra as diversas bibliotecas de uma instituição, com o objetivo de otimizar o acesso à informação e apoiar as atividades acadêmicas, científicas e culturais. Esse sistema utiliza tecnologias e ferramentas que possibilitam a automação e a centralização dos processos bibliotecários, aprimorando a gestão do acervo e os serviços prestados aos usuários.

Um componente essencial do SiBi é o uso de sistemas automatizados de gestão bibliotecária, que facilitam o controle e a administração das operações diárias, como empréstimos, renovações, reservas e consultas ao acervo. Essas funcionalidades são fundamentais para garantir um atendimento eficiente e para possibilitar o acesso remoto e simplificado ao acervo, seja ele físico ou digital. Dentre as principais funções desses sistemas automatizados destacam-se:

- **Empréstimos e Devoluções:** Registro detalhado dos materiais emprestados, controle dos prazos para devolução e acompanhamento do histórico de uso dos usuários.
- **Renovações:** Permite a extensão do prazo de devolução, desde que não haja reservas pendentes, facilitando a continuidade do acesso ao material.
- **Reservas:** Possibilita aos usuários solicitarem livros ou outros documentos que estejam indisponíveis no momento, organizando a fila de espera de forma automática.
- **Consulta ao Catálogo:** Disponibiliza um ambiente virtual para pesquisa do acervo, facilitando a localização e a seleção dos materiais disponíveis.

2.2 Sistemas de Informação Aplicados a Bibliotecas

Neste tópico será discutido o funcionamento de um sistema bibliotecário, entendido como um conjunto de ferramentas que automatizam processos como catalogação, circulação, empréstimos e gestão de usuários. Serão também apresentados conceitos e aplicações de sistemas web e sistemas mobile nessa área. Essas tecnologias complementam os sistemas bibliotecários tradicionais, ampliando o alcance e a acessibilidade dos serviços, possibilitando consultas, contribuindo para a transformação das bibliotecas em ambientes digitais e interativos, alinhados às demandas contemporâneas.

2.2.1 Sistema bibliotecário

Os sistemas bibliotecários podem ser compreendidos como um conjunto de ferramentas tecnológicas e processos que auxiliam na gestão e organização das atividades de uma biblioteca. Os sistemas de automação de bibliotecas são bases de dados projetadas para controlar as principais funções de uma biblioteca, como aquisição, catalogação, circulação e administração do acervo, otimizando o acesso à informação e o controle de empréstimos. Esses sistemas possibilitam o gerenciamento eficiente dos recursos informacionais e a integração entre diferentes módulos de serviço, promovendo maior agilidade e precisão nas operações (RIBEIRO; SILVA, 2019).

Os sistemas automatizados permitem não apenas registrar e controlar empréstimos de forma precisa, como também oferecer funcionalidades como alertas de devolução, bloqueio automático de usuários inadimplentes, emissão de relatórios estatísticos e integração com catálogos online.

Um sistema de biblioteca bem estruturado deve priorizar a organização do catálogo de forma a facilitar a busca por obras, autores ou gêneros, o que também se aplica aos sistemas de empréstimo.

2.2.2 Sistema Web

Com o surgimento e a popularização da internet, o uso de ambientes digitais tornou-se cada vez mais presente na vida cotidiana dos indivíduos. Entre essas tecnologias, destaca-se a World Wide Web (WWW), criada em 1989, que ao longo das décadas consolidou-se como a principal plataforma para comunicação, informação, entretenimento e transações comerciais. O crescimento contínuo do acesso à rede permitiu não apenas a expansão de serviços online, mas também a transformação da forma como pessoas e instituições interagem.

No cenário brasileiro, a internet alcançou proporções expressivas. Em 2024, aproximadamente 93,6% dos domicílios possuíam acesso à rede, enquanto 88% da população com 10 anos ou mais fazia uso regular da internet (IBGE, 2024). Assim, a possibilidade de acesso remoto a catálogos, acervos digitais e serviços de gestão tornou-se cada vez mais relevante.

No entanto, estudos revelam que apenas 9% das bibliotecas brasileiras possuem site próprio, enquanto a maioria ainda depende de páginas institucionais de secretarias ou prefeituras para manter alguma presença digital (ONU Brasil, 2019). Esse dado demonstra que, apesar da crescente importância da web, muitas bibliotecas ainda não exploram plenamente o potencial oferecido por sistemas online.

2.2.3 Sistema Mobile

O termo *mobile* refere-se a tecnologias móveis, ou seja, aquelas que permitem o uso de dispositivos sem fio com mobilidade, como celulares, smartphones, tablets e outros equipamentos portáteis. A ideia de sistemas mobile está atrelada à expansão das Tecnologias da Informação e Comunicações, conhecido como TICs, um conjunto de recursos tecnológicos para obter, processar e gerar informações que são tornadas acessíveis por meio de redes de comunicação, se popularizou nos telefones no final dos anos 1990 e do surgimento dos smartphones no início dos anos 2000 (PUCPR, 2024).

A aplicação de sistemas mobile em bibliotecas surge como resposta à necessidade de modernização e ao comportamento do usuário contemporâneo, que valoriza a autonomia, a agilidade e a disponibilidade remota de serviços. Bibliotecas que integram sistemas móveis oferecem funcionalidades como: consulta ao catálogo, renovação e reserva de materiais, acesso a e-books, visualização de histórico de

empréstimos, notificações de devolução, autenticação de usuário e até auto empréstimo por QR Code. Tais funcionalidades podem ser acessadas via aplicativos próprios ou interfaces web responsivas.

2.3 Trabalhos Correlatos

Durante o processo de desenvolvimento do projeto, é essencial realizar uma análise criteriosa dos sistemas existentes (Sistema SophiA e Koha) e das abordagens já documentadas que tratam de problemas semelhantes ao objeto de estudo (Pergamum e Tanca). Essa análise, fundamentada em trabalhos correlatos, permite compreender o panorama atual do projeto, identificar lacunas e avaliar metodologias e tecnologias já empregadas. Portanto, os principais sistemas serão discutidos nas seções subsequentes, permitindo uma análise crítica e comparativa que fundamenta as escolhas e direções adotadas neste projeto.

Figura 1 – Sistemas de referência



Fonte: Elaborado pelos autores (2025) | Canva

2.3.1 Sistema Biblioteca SophiA

O Sistema Biblioteca SophiA é um software de gestão pago, criado pela Prima para a gestão de bibliotecas acadêmicas, baseada nos padrões internacionais de catalogação e comunicação de dados. Através do sistema, os alunos, professores e demais usuários podem fazer a renovação do seu material, bem como reservar livros,

fazer consulta bibliográfica, acompanhar o histórico de empréstimo e fazer a troca da senha (SOPHIA, s.d.).

A adoção de sistemas automatizados de gestão bibliotecária representa um avanço significativo para instituições acadêmicas, pois otimiza processos internos, melhora o atendimento ao usuário e contribui para a modernização dos serviços de informação. Esses sistemas permitem não apenas o controle eficiente de acervos físicos e digitais, mas também a integração entre unidades, a ampliação do acesso remoto e o suporte a decisões administrativas por meio de relatórios e indicadores. Instituições que implementam soluções como o SophiA Biblioteca frequentemente observam melhorias tanto na produtividade dos profissionais quanto na experiência dos usuários. Um exemplo claro disso é relatado por Gilmar Vicente, bibliotecário e membro do Grupo Gestor do SophiA na Universidade Estadual de Campinas (Unicamp), ao afirmar:

Após doze meses de utilização do SophiA Biblioteca na Unicamp as avaliações de produtividade nas atividades de catalogação aumentaram significativamente e o tempo de atendimento aos usuários na circulação diminuíram. Podemos considerar que integração seria a palavra-chave das atividades executadas pelo SophiA no sistema de Bibliotecas da Unicamp. Trata-se de um produto atual, versátil. Com a versão mobile e o terminal web a utilização dos usuários também é facilitada. As atividades relativas à biblioteca (renovação, reservas e recentemente, chat e endereços das bibliotecas do SBU) podem ser operacionalizadas facilmente. (VICENTE, [s.d.] apud SOPHIA, 2025)

2.3.2 Sistema Koha

O Koha é o software livre e gratuito para gestão de bibliotecas mais utilizado no mundo e implantado em milhares de bibliotecas de todos os continentes. O sistema foi desenvolvido pela Katipo Communications para a Horowhenua Library Trust, da Nova Zelândia. Significa “presente” ou “doação” em maori, língua dos aborígenes neozelandeses (KOHA COMMUNITY, s.d.). Com isso, seus desenvolvedores ofertavam ao mundo uma ferramenta para automação de bibliotecas que logo foi adotada em todo o mundo. Atualmente é um software moderno e robusto, cuja função é informatizar bibliotecas de todos os tipos, pois atende a todos os processos da biblioteca com utilização de padrões internacionais.

O acesso à informação desempenha um papel fundamental na sociedade contemporânea, especialmente em ambientes educacionais. Nesse cenário, o uso de tecnologias de gestão bibliotecária se mostra essencial para garantir a eficiência dos serviços prestados. O Instituto Federal de Educação, Ciência e Tecnologia (IFSP) adotou o sistema Koha como solução para a gestão integrada de suas bibliotecas, destacando-se por sua abrangência e funcionalidade em rede.

Um exemplo relevante sobre a modernização promovida pelo Sistema Koha é apresentado por Fabiana Lopes do Nascimento, do Instituto Federal da Paraíba (IFPB), durante a SNBU 2023 (Seminário Nacional de Bibliotecas Universitárias). A autora relata:

A implantação do Sistema Koha é multicampi, integrada em rede e opera atendendo a toda a comunidade do IFPB, podendo o usuário fazer consulta, reserva e renovações online. O Sistema de Gestão Koha nos permite oferecer resultados mais hábeis, em favor de mudanças nas atividades da biblioteca.
(NASCIMENTO; NASCIMENTO, 2023)

2.3.3 *Touch System Tanca*

Um sistema Touch System Tanca serve para automatizar e otimizar processos em estabelecimentos comerciais, oferecendo uma interface de usuário mais rápida, intuitiva e eficiente através de uma tela sensível ao toque, dispensando o uso de teclado e mouse, o que facilita o atendimento ao cliente e a gestão do negócio (TANCA, s.d.). Ou seja, ele não cria um sistema do zero, mas oferece uma plataforma pronta que pode ser implementada e personalizada conforme as necessidades de cada organização.

A implementação do sistema envolve a instalação do software e a configuração de suas funcionalidades conforme as especificidades da instituição. Após uma análise detalhada realizada por nós, desenvolvedores do projeto e alunos do SESI e do curso SENAI, constatamos que, nas bibliotecas das escolas SESI, o Tanca tem sido efetivamente utilizado para gerenciar o acervo, registrar empréstimos e devoluções, além de consultas ao catálogo.

O Touch System Tanca é projetado para ser fácil de usar, com uma interface que dispensa o uso de teclado e mouse, sendo operado diretamente por uma tela sensível ao toque que permite os bibliotecários e alunos operem diretamente na ecrã

tátil. Isso não apenas otimiza o tempo de atendimento ao público, mas também melhora a experiência do usuário ao tornar o processo de consulta e devolução de materiais mais ágil e sem complicações.

2.3.4 *Pergamum*

O Sistema Integrado de Bibliotecas Pergamum é um software proprietário amplamente utilizado, sendo uma das principais soluções pagas disponíveis no mercado para a gestão de bibliotecas (Pergamum, s.d.). O desenvolvimento do sistema Pergamum foi realizado por alunos da área de computação, em colaboração com bibliotecários da Pontifícia Universidade Católica do Paraná (PUCPR), o que assegurou a adequação do software às necessidades específicas da gestão bibliotecária.

O sistema Pergamum é amplamente utilizado por diversas instituições de ensino brasileiras, como o SENAI, a Universidade Presbiteriana Mackenzie, o Centro Universitário São Camilo, entre outras, para a gestão de bibliotecas acadêmicas. A plataforma permite que os usuários realizem consultas ao acervo de forma online, visualizando informações detalhadas como a quantidade de exemplares disponíveis, a localização física dos itens e a disponibilidade para empréstimo.

2.4 Conceitos de Modelagem do Sistema

Este tópico aborda os conceitos teóricos de modelagem de sistemas, com foco no Diagrama de Casos de Uso e no Diagrama Entidade-Relacionamento (DER). Nele, são explicados os elementos fundamentais de cada diagrama, como atores, casos de uso, entidades, relacionamentos e cardinalidades, além de suas aplicações no contexto de desenvolvimento de software. A base teórica apresentada serve como referência para a compreensão das técnicas utilizadas na modelagem prática do sistema de gerenciamento de bibliotecas, garantindo alinhamento com as melhores práticas de engenharia de software e facilitando a comunicação entre a equipe de desenvolvimento e os stakeholders.

2.4.1 Diagrama de caso de uso

De acordo com informações da Lucidchart (2024), o diagrama de caso de uso resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema. Para criar um, são usados um conjunto de símbolos e conectores especializados. Um bom diagrama de caso de uso ajuda sua equipe a representar e discutir:

- Cenários em que o sistema ou aplicativo interage com pessoas, organizações ou sistemas externos;
- Metas que o sistema ou aplicativo ajuda essas entidades (conhecidas como atores) a atingir;
- O escopo do sistema.

O diagrama de caso de uso não oferece muitos detalhes — não mostra a ordem em que os passos são executados. Em vez disso, um diagrama de caso de uso adequado dá uma visão geral do relacionamento entre casos de uso, atores e sistemas (LUCIDCHART, 2024).

2.4.2 Diagrama entidade-relacionamento (DER)

Um diagrama entidade relacionamento (ER) é um tipo de fluxograma que ilustra como “entidades”, por exemplo: pessoas, objetos ou conceitos, se relacionam entre si dentro de um sistema. Diagramas ER são mais utilizados para projetar ou depurar bancos de dados relacionais nas áreas de engenharia de software, sistemas de informações empresariais, educação e pesquisa. Estão relacionados com diagramas de estrutura de dados (DEDs), que incidem sobre as relações de elementos dentro de entidades em vez de relações entre as próprias entidades (LUCIDCHART, 2024).

Também conhecidos como DERs, ou modelos ER, usam um conjunto definido de símbolos, tais como retângulos, diamantes, ovais e linhas de conexão para representar a interconectividade de entidades, relacionamentos e seus atributos. Eles espelham estruturas gramaticais, onde entidades são substantivos e relacionamentos são verbos.

COMPONENTES E FUNCIONALIDADES DE UM DIAGRAMA ER

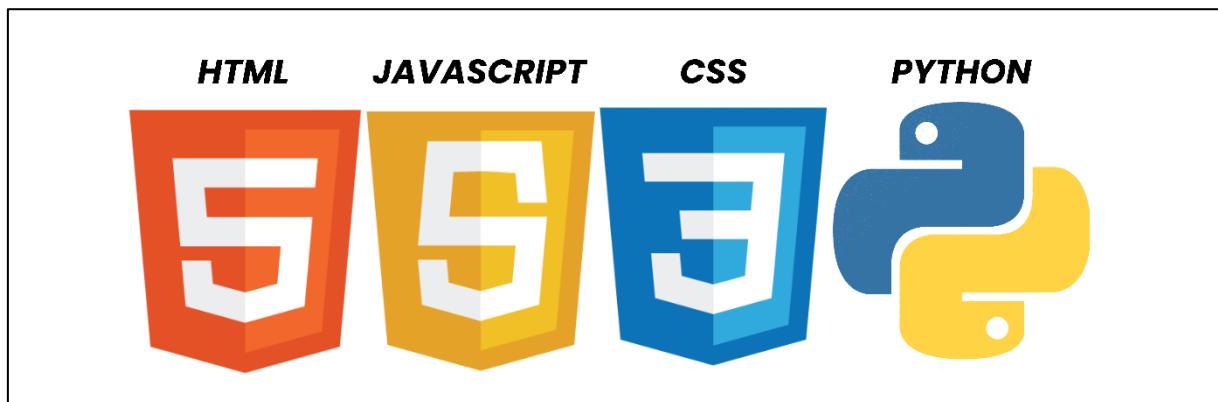
Diagramas ER são compostos por:

- **Entidade:** Algo que pode ser definido e que pode ter dados armazenados sobre ele — como uma pessoa, um objeto, conceito ou evento. Pense em entidades como substantivos. Normalmente representado como um retângulo.
- **Relacionamento:** Como entidades atuam umas sobre as outras ou estão associadas uma com a outra. Pense em relacionamentos como verbos. Relacionamentos são tipicamente representados por diamantes ou etiquetas diretamente sobre as linhas de ligação.
- **Atributo:** A propriedade ou característica de uma entidade, muitas vezes representada por um oval ou círculo.
- **Cardinalidade:** Define os atributos numéricos da relação entre duas entidades ou conjuntos de entidades. Os três principais relacionamentos cardinais são um-para-um, um-para-muitos e muitos-para-muitos.

2.5 Ferramentas utilizadas

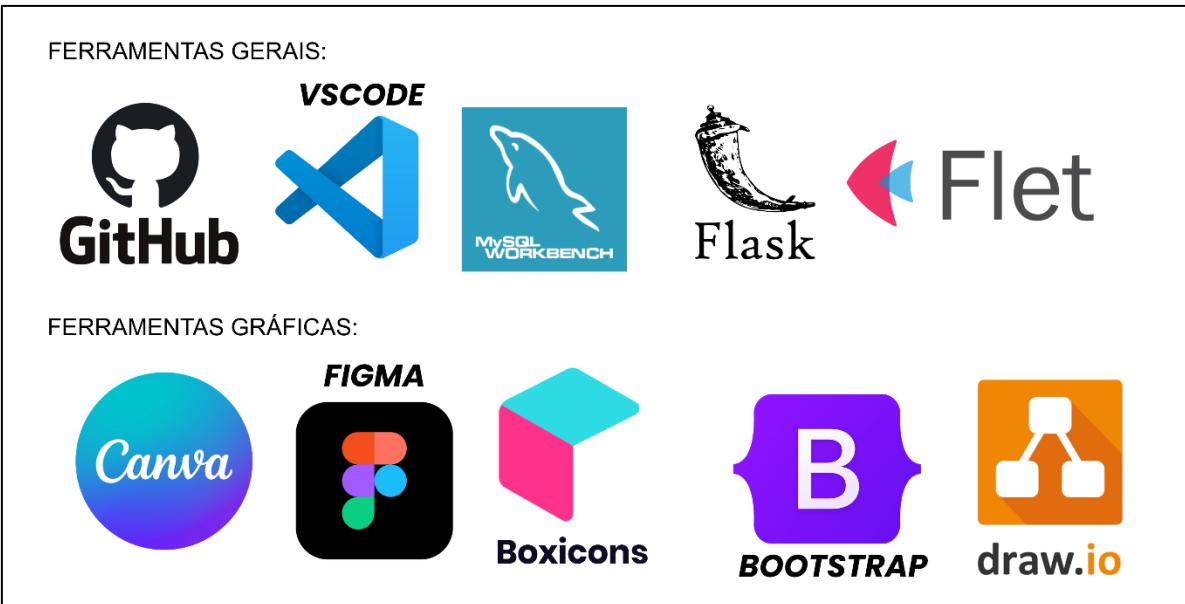
Esta seção elenca as linguagens de programação e descreve as demais ferramentas e tecnologias adotadas durante a elaboração do projeto.

Figura 2 – Linguagens utilizadas



Fonte: Elaborado pelos autores (2025) | Canva

Figura 3 – Ferramentas utilizadas



Fonte: Elaborado pelos autores (2025) | Canva

2.5.1 *Linguagens*

As linguagens de marcação (como HTML ou XML) são sistemas que usam símbolos ou “tags” inseridas no conteúdo para indicar sua estrutura ou formatação, sem incorporar lógica de execução. Já as linguagens de estilo (por exemplo, CSS) focalizam a apresentação visual de documentos – fontes, cores, layout – aplicadas sobre uma estrutura já existente. Em contraste, as linguagens de programação (como Python ou Javascript) são formalmente definidas para dar instruções a um computador e controlar processos: elas incluem sintaxe, semântica e regras para execução de tarefas.

2.5.1.1 *HTML*

O HTML (HyperText Markup Language) é a linguagem base utilizada para estruturar conteúdos em páginas web. Ela define a organização dos elementos da interface, como textos, imagens, vídeos, formulários e links, por meio de tags que indicam o significado e a hierarquia de cada componente (ALURA, 2023).

No projeto, o HTML foi empregado para construir a estrutura semântica da aplicação, garantindo que cada parte da interface tivesse uma função bem definida.

Essa organização facilita a acessibilidade, manutenção e compatibilidade com diferentes navegadores e dispositivos.

Exemplo 1: Estrutura básica de uma página HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Página Exemplo</title>
</head>
<body>
    <h1>Bem-vindo ao nosso site</h1>
    <p>Este é um exemplo de estrutura HTML simples.</p>
</body>
</html>
```

Fonte: Elaborado pelos autores (2025)

Neste exemplo, o elemento `<!DOCTYPE html>` informa ao navegador que o documento está escrito em HTML5. A tag `<html>` delimita o início e o fim do código HTML, enquanto o atributo `lang="pt-br"` indica o idioma da página. Dentro da seção `<head>`, a tag `<meta charset="UTF-8">` define a codificação de caracteres para suportar acentuação, e `<title>` especifica o título que será exibido na aba do navegador. Já no `<body>`, o `<h1>` representa um título principal e `<p>` indica um parágrafo de texto.

Exemplo 2: Formulário simples

```
<form action="/enviar" method="POST">
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome">
    <button type="submit">Enviar</button>
</form>
```

Fonte: Elaborado pelos autores (2025)

Neste caso, a tag <form> cria um formulário e define o destino dos dados com o atributo action, enquanto method="POST" determina o método de envio das informações. A tag <label> associa um texto descritivo a um campo de entrada, ligado ao atributo for. Já <input> cria o campo onde o usuário digita seu nome, e <button> adiciona um botão de envio. Essa estrutura foi utilizada no projeto para coleta de dados de usuários e validação posterior via back-end.

2.5.1.2 CSS

O CSS (Cascading Style Sheets) é uma linguagem responsável pela definição visual e estilização das páginas web. Ela controla aspectos como cores, fontes, margens, espaçamento, alinhamento e responsividade, permitindo a separação entre estrutura (HTML) e estilo (CSS) (ALURA, 2021).

No projeto, o CSS foi utilizado para criar uma identidade visual coerente e agradável, aplicando paletas de cores, tipografia padronizada e layout responsivo. Além disso, o uso de media queries possibilitou que a interface se adaptasse a diferentes tamanhos de tela, aprimorando a usabilidade em dispositivos móveis.

Exemplo 1: Gradiente de fundo para a página

```
body {  
    font-family: Arial, sans-serif;  
    color: #fff;  
    background: linear-gradient(135deg, #1e73be, #ff7a59);  
    min-height: 100vh;  
    margin: 0;  
    padding: 0;  
}
```

Fonte: Elaborado pelos autores (2025)

Neste exemplo, o seletor body aplica estilos ao corpo da página. O font-family define a tipografia utilizada, garantindo legibilidade. A cor do texto é definida como branca (color: #fff) para contrastar com o fundo. A propriedade background: linear-

`gradient(135deg, #1e73be, #ff7a59)` cria um gradiente linear que transita entre duas cores — azul (`#1e73be`) e laranja (`#ff7a59`) — em um ângulo de 135 graus. Isso proporciona um visual moderno e atraente, dando profundidade à página. As propriedades `min-height: 100vh`, `margin: 0` e `padding: 0` asseguram que o gradiente ocupe toda a altura da tela sem margens indesejadas.

Exemplo 2: Responsividade com media queries e estilização de botão

```
button {  
    background: green;  
    color: white;  
    border: none;  
    padding: 12px 24px;  
    border-radius: 8px;  
    font-size: 16px;  
    cursor: pointer;  
}  
  
@media (max-width: 768px) {  
    button {  
        width: 100%;  
        font-size: 14px;  
        padding: 10px 20px;  
    }  
}
```

Fonte: Elaborado pelos autores (2025)

Neste exemplo, o seletor `button` estiliza todos os botões da página. O `color: white` define a cor do texto como branco. O botão recebe bordas arredondadas (`border-radius: 8px`), espaçamento interno (`padding`), tamanho de fonte (`font-size`) e um cursor interativo (`cursor: pointer`).

A media query `@media (max-width: 768px)` aplica estilos específicos para telas com largura máxima de 768 pixels, geralmente tablets e smartphones. Dentro dela, o botão ocupa 100% da largura disponível, reduz o tamanho da fonte e ajusta o padding,

garantindo que os elementos continuem proporcionais e acessíveis em dispositivos móveis. Isso demonstra o conceito de design responsivo, essencial para interfaces modernas que se adaptam a diferentes resoluções de tela.

2.5.1.3 *JavaScript*

O JavaScript é uma linguagem de programação interpretada que adiciona interatividade e dinamismo às páginas web. Com ela, é possível manipular o DOM (Document Object Model), validar formulários, criar animações e responder a eventos do usuário em tempo real (ALURA, 2021).

No contexto do projeto, o JavaScript foi empregado para implementar funcionalidades interativas, como validação de dados antes do envio ao servidor, efeitos visuais e atualização dinâmica de elementos. Essa camada de interação complementou o HTML e o CSS, tornando a aplicação mais funcional e atrativa.

Exemplo 1: Exibição de mensagem ao clicar em um botão

```
function mostrarMensagem() {  
    alert("Bem-vindo ao nosso site!");  
}
```

Fonte: Elaborado pelos autores (2025)

Nesse exemplo, a função `mostrarMensagem()` utiliza o método `alert()` para exibir uma mensagem pop-up na tela. Essa função pode ser acionada, por exemplo, através de um evento `onclick` em um botão HTML. Esse tipo de interação foi aplicado no projeto para confirmar ações do usuário, como o envio de formulários.

Exemplo 2: Validação simples de formulário

```
function validarFormulario() {  
    const nome = document.getElementById("nome").value;  
    if (nome === "") {  
        alert("Por favor, preencha o campo nome.");  
        return false;  
    }  
}
```

```
    return true;  
}
```

Fonte: Elaborado pelos autores (2025)

Aqui, a função validarFormulario() obtém o valor do campo de entrada nome através do método document.getElementById(). Em seguida, a estrutura condicional if verifica se o campo está vazio e, caso esteja, exibe um alerta e impede o envio retornando false. Caso contrário, a função retorna true, permitindo o envio. Essa lógica foi aplicada para prevenir erros de preenchimento e melhorar a usabilidade da aplicação.

2.5.1.4 Python

O Python é uma linguagem de programação de alto nível, caracterizada pela legibilidade e simplicidade de sintaxe, amplamente utilizada em desenvolvimento web, automação, ciência de dados e inteligência artificial (HABBEMA, 2023).

No projeto, o Python foi utilizado no desenvolvimento do back-end, com o auxílio do framework Flask, para lidar com requisições HTTP, gerenciamento de rotas, processamento de dados e integração com o banco de dados MySQL. Sua versatilidade e ampla base de bibliotecas facilitaram a implementação de funcionalidades do servidor e a comunicação com o front-end.

Exemplo 1: Rota simples com Flask

```
from flask import Flask  
  
app = Flask(__name__)  
  
@app.route('/')  
def home():  
    return "Bem-vindo à aplicação!"
```

Fonte: Elaborado pelos autores (2025)

Neste exemplo, o módulo Flask é importado e instanciado como app, criando o servidor da aplicação. O decorador @app.route('/') define a rota principal (raiz do site), enquanto a função home() retorna o texto exibido quando o usuário acessa essa rota. Essa estrutura básica foi utilizada no projeto para criar diferentes páginas e gerenciar a comunicação entre front-end e back-end.

Exemplo 2: Manipulação básica de dados

```
dados = ["Produto A", "Produto B", "Produto C"]
for item in dados:
    print(f"Item disponível: {item}")
```

Fonte: Elaborado pelos autores (2025)

Nesse trecho, uma lista chamada dados armazena três elementos de texto. O laço for percorre cada item da lista e exibe uma mensagem formatada com print(). O uso da **f-string** (`f"texto {variável}"`) permite inserir variáveis dentro de uma string de maneira mais legível. Estruturas semelhantes foram aplicadas no projeto para organizar informações provenientes do banco de dados antes de exibi-las na interface.

2.5.2 GitHub

O GitHub é um repositório de hospedagem de serviços Git, todavia possui outras tantas características próprias. As interações entre os seus usuários são de natureza complexa e ocorrem em diferentes formas. Principalmente porque pode ser considerado como uma rede social além de uma plataforma colaborativa. Usando o GitHub, os programadores podem interagir e colaborar em torno de repositórios de código aberto, o que permite que eles façam download, cooperem, compartilhem além de outras funcionalidades.

Este repositório foi escolhido para acrescentar ao nosso projeto pela facilidade em criar portfólios públicos que podem ser úteis para procurar emprego, colaborar em projetos com outros desenvolvedores e compartilhar códigos com mais facilidade.

2.5.3 Visual Studio Code

O Visual Studio Code é um editor de código-fonte criado pela Microsoft com o objetivo de auxiliar programadores na criação de códigos de softwares. Logo, ele é muito utilizado pelos desenvolvedores para escrever, editar e gerenciar os códigos que estão desenvolvendo, especialmente nas fases de codificação e de testes. O VS Code suporta diversas linguagens de programação e oferece extensões, depuração integrada, controle de versões e personalização do ambiente.

Durante o desenvolvimento, o VS Code foi a ferramenta principal para escrita, edição e testes do código. Utilizamos este editor por sua facilidade na construção de códigos e por sua interface limpa, personalizável e organizada.

2.5.4 MySQL Workbench

O MySQL Workbench é uma ferramenta visual oficial para o gerenciamento e modelagem de bancos de dados MySQL. Ela permite a criação, edição e execução de consultas SQL, além de possibilitar o design de modelos de dados de forma gráfica (ESCOLA DNC, 2025).

No projeto, o MySQL Workbench foi utilizado para o desenvolvimento e administração do banco de dados, facilitando a criação das tabelas, relações e consultas necessárias para o funcionamento do sistema, também para a criação do diagrama de entidade-relacionamento (DER). Sua interface intuitiva e os recursos integrados de modelagem contribuíram para uma melhor organização e eficiência no processo de desenvolvimento.

2.5.5 Flask

O Flask é um microframework em Python voltado ao desenvolvimento de aplicações web. Ele fornece os recursos essenciais para criação de rotas, tratamento de requisições HTTP, autenticação e integração com bancos de dados, mantendo uma estrutura leve e flexível (ALURA, s.d.).

No projeto, o Flask foi responsável por gerenciar o back-end da aplicação, processando dados enviados pelo front-end e retornando respostas dinâmicas. Sua

simplicidade e compatibilidade com bibliotecas externas facilitaram a implementação das funcionalidades do sistema e o desenvolvimento ágil.

2.5.6 *Flet*

O Flet é um framework moderno para criação de interfaces gráficas em Python, permitindo o desenvolvimento de aplicações interativas e responsivas de forma simples e intuitiva. Sua proposta é possibilitar que desenvolvedores criem interfaces ricas sem necessidade de conhecimentos avançados em HTML, CSS ou JavaScript, já que toda a lógica visual é construída diretamente em Python (FLET, s.d.).

No projeto, o Flet foi utilizado para desenvolver o chat de atendimento rápido do Sistema Integrado de Bibliotecas (SiBi), responsável por oferecer respostas automáticas sobre horários, unidades e procedimentos de recuperação de senha. A biblioteca permitiu criar uma interface amigável, responsiva e com elementos visuais como balões de conversa, botões de sugestão e animação de digitação, proporcionando uma experiência semelhante a aplicativos modernos de mensagens.

Além disso, uma das vantagens do Flet é sua capacidade de gerar aplicações que podem ser executadas tanto em desktops quanto em dispositivos móveis. Dessa forma, o chat do SiBi também pode ser acessado e utilizado diretamente pelo celular, mantendo sua organização visual e usabilidade mesmo em telas menores. O Flet facilitou ainda o gerenciamento de eventos, controle de interface e atualização dinâmica dos componentes visuais, permitindo que o assistente respondesse em tempo real às interações do usuário.

Exemplo 1: Estrutura básica da aplicação com Flet

```
import flet as ft

def main(page: ft.Page):
    page.title = "Sibi - Atendimento"
    page bgcolor = "white"
    chat = ft.Column(expand=True)
    page.add(chat)
ft.app(main)
```

Fonte: Elaborado pelos autores (2025)

Nesta estrutura inicial, o Flet é importado e executado por meio da função ft.app(main). A função main() recebe o objeto page, responsável por gerenciar todos os elementos da interface.

O componente ft.Column() organiza os itens verticalmente, servindo como estrutura base do chat. Essa base foi expandida ao longo do projeto para acomodar balões de texto, botões, imagens e outros elementos de interação.

Exemplo 2: Respostas automáticas do assistente

```
def obter_resposta (texto):
    t = texto.lower()
    if "horário" in t:
        return "Horários de funcionamento: 08h às 17h."
    if "unidade" in t:
        return "Unidades disponíveis: 265, 094, 166 e 221."
    return "Desculpe, não consegui entender. Pode tentar novamente?"
```

Fonte: Elaborado pelos autores (2025)

Nesse trecho, a função obter_resposta() recebe a mensagem do usuário e realiza verificações baseadas em palavras-chave. Cada condição retorna uma resposta específica, posteriormente exibida no balão do assistente. Estruturas semelhantes foram utilizadas no projeto final para organizar o conteúdo exibido ao usuário.

2.5.7 Ferramentas gráficas

As ferramentas gráficas desempenham um papel central no desenvolvimento de interfaces modernas, contribuindo tanto para a identidade visual quanto para a experiência do usuário.

No contexto deste projeto, diferentes recursos foram aplicados de forma complementar para garantir um design consistente, funcional e visualmente atraente.

Plataformas como Canva e Figma possibilitaram desde a criação da identidade visual até o planejamento detalhado da interface, enquanto bibliotecas e frameworks como BoxIcons e Bootstrap ofereceram suporte técnico para compor elementos gráficos e assegurar responsividade. A integração dessas ferramentas permitiu alinhar estética, usabilidade e eficiência no processo de desenvolvimento, resultando em uma aplicação visualmente coesa e intuitiva.

2.5.7.1 *Canva*

O Canva é uma plataforma acessível e repleta de recursos de design gráfico que possibilita a criação de materiais visuais de forma colaborativa e intuitiva. Ela oferece modelos prontos, elementos gráficos, fontes e paletas de cores, sendo ideal para criação de logotipos, banners e layouts (JR. COMUNICAÇÃO VISUAL, 2025). No projeto, o Canva foi utilizado para desenvolver a identidade visual da aplicação, incluindo o logotipo e os elementos gráficos da interface.

2.5.7.2 *Figma*

O Figma é uma ferramenta de prototipagem e design de interfaces (UI/UX) baseada na web, amplamente usada para criar wireframes, mockups e protótipos interativos. Ele permite o trabalho colaborativo em tempo real, o que facilita o desenvolvimento em equipe (ALURA, 2025).

Durante o projeto, o Figma foi essencial para o planejamento visual da aplicação, permitindo visualizar o fluxo de navegação e testar a disposição dos elementos antes da implementação. Assim, a equipe pôde alinhar o design com a usabilidade, garantindo uma experiência de usuário fluida e intuitiva.

2.5.7.3 *BoxIcons*

O Boxicons é uma biblioteca de ícones vetoriais de código aberto desenvolvida para uso em interfaces web. Sua principal característica é a facilidade de integração com projetos front-end por meio de classes CSS simples, permitindo adicionar ícones de forma leve e personalizável.

No projeto, a ferramenta Boxicons foi utilizada para compor elementos visuais da interface, como botões, menus e indicadores, proporcionando um design mais moderno e intuitivo. Sua ampla variedade de ícones e compatibilidade com diferentes navegadores contribuíram para uma melhor experiência visual do usuário.

2.5.7.4 *Bootstrap*

O Bootstrap é um framework front-end baseado em HTML, CSS e JavaScript, criado para acelerar o desenvolvimento de interfaces responsivas. Ele oferece componentes pré-estilizados, como botões, formulários, modais e sistemas de grid (ALURA, 2021).

No projeto, o Bootstrap foi empregado para garantir responsividade e padronização visual, adaptando automaticamente a interface a diferentes resoluções de tela. Sua utilização permitiu uma melhor experiência do usuário e reduziu o tempo de desenvolvimento, mantendo coerência estética em toda a aplicação.

2.5.7.5 *Draw.io*

O Draw.io é um conjunto de tecnologias para a criação de aplicativos de diagramação e o software de diagramação para usuários finais baseado em navegador mais utilizado no mundo (DRAW.IO, s.d.).

No desenvolvimento do projeto foi utilizado para a elaboração do diagrama de Casos de Uso, utilizando a ferramenta da UML (Unified Modeling Language) que permite representar as funcionalidades do sistema sob a perspectiva de quem o utiliza.

2.6 Ferramentas futuras

Com o objetivo de aprimorar o sistema escolar bibliotecário e torná-lo mais dinâmico e completo, considera-se a adoção de novas ferramentas tecnológicas em futuras versões do projeto. Entre as possibilidades, destaca-se a integração de agentes de Inteligência Artificial, que poderão automatizar tarefas rotineiras, como recomendações de livros e organização de registros, utilizando plataformas como o n8n para facilitar a automação e o gerenciamento dos fluxos de dados.

Além disso, planeja-se a implementação de um framework mobile, como o Flutter ou o Flet, visando ampliar o acesso ao sistema por meio de dispositivos móveis. Essa adaptação permitiria que alunos, professores e bibliotecários utilizassem as funcionalidades de forma prática e intuitiva, fortalecendo a acessibilidade e modernizando a experiência do usuário.

2.6.1 Agente de IA com *n8n*

Entre as principais possibilidades de evolução estão a integração de agentes de Inteligência Artificial e a utilização de frameworks mobile para expansão multiplataforma. Um agente de Inteligência Artificial (IA) é um sistema autônomo capaz de analisar informações, tomar decisões estratégicas e executar tarefas complexas com pouca ou nenhuma intervenção humana (AWS, 2024). Diferente de sistemas tradicionais que seguem fluxos fixos e limitados, os agentes de IA possuem uma estrutura mais dinâmica, sendo capazes de processar dados em tempo real, aprender com interações e utilizar ferramentas externas para gerar soluções inteligentes. Pode-se compará-los a um “cérebro digital”, que interpreta contextos, acessa diferentes fontes de informação e age de forma coordenada e precisa.

No contexto deste projeto, a integração desses agentes poderá ser viabilizada por meio do *n8n*, uma plataforma de automação e integração visual que permite conectar diversos serviços e executar fluxos personalizados sem necessidade de programação avançada. O *n8n* conta com centenas de integrações nativas e permite criar automações complexas por meio de gatilhos (Trigger), modelos de linguagem (LLM) e ferramentas auxiliares, como planilhas online, e-mails e APIs externas. A utilização dessa tecnologia possibilitaria que o sistema bibliotecário automatizasse tarefas rotineiras — como criação de chatbots inteligentes integrados ao sistema bibliotecário —, tornando o ambiente escolar mais eficiente e interativo.

2.6.2 Framework de desenvolvimento mobile

Com o objetivo de tornar o sistema escolar bibliotecário mais acessível, moderno e compatível com diferentes dispositivos, propõe-se a futura integração com frameworks voltados ao desenvolvimento mobile, como Flutter e Flet.

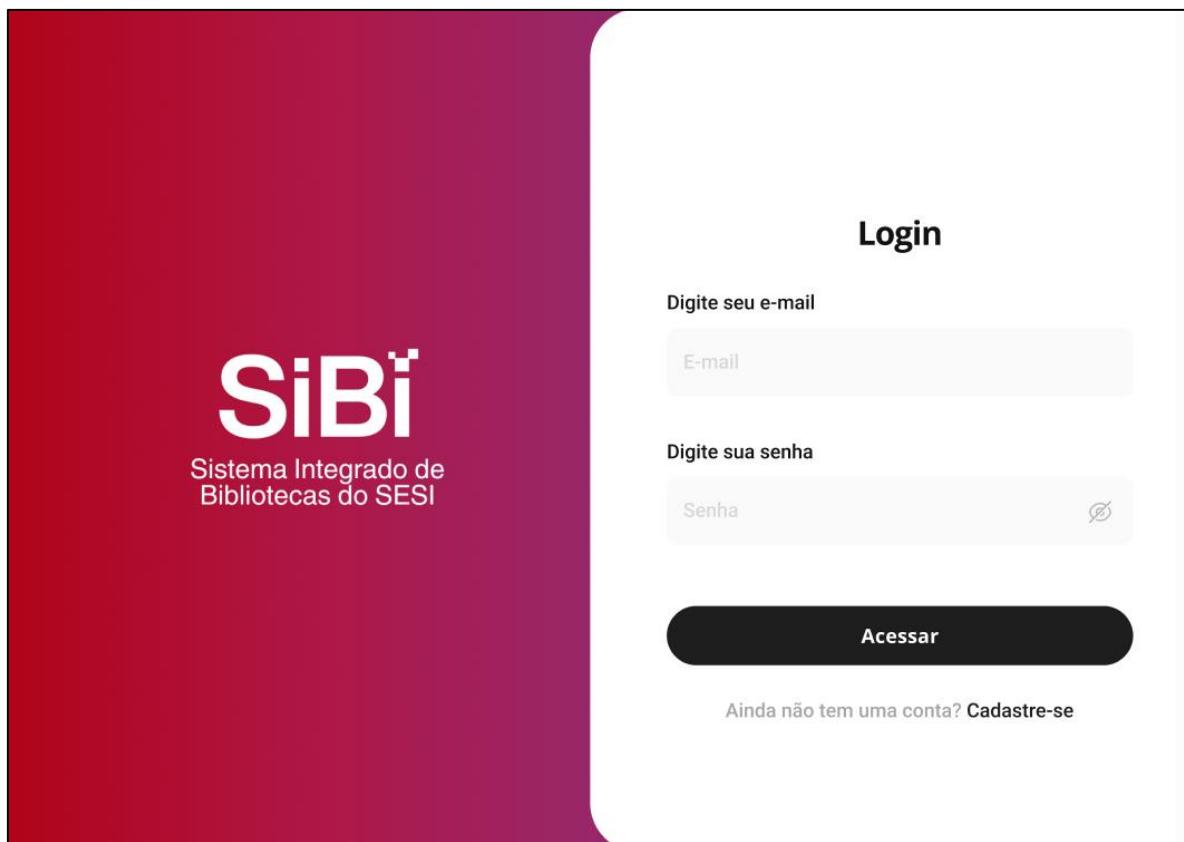
O Flutter, desenvolvido pela Google, é um framework de código aberto utilizado para criar aplicações nativas multiplataforma a partir de um único código-fonte. Sua estrutura é composta por três camadas principais: o framework (onde se desenvolvem as interfaces com a linguagem Dart), a engine – C/C++ (que renderiza os elementos visuais e executa o código de forma otimizada) e o embedder – específico da plataforma (responsável por adaptar o aplicativo a cada sistema operacional) (ALURA, 2025). Permite desenvolver funcionalidades como consulta de acervo, reserva e renovação de empréstimos, notificações de devolução e recomendações automáticas de leitura poderiam ser facilmente integradas ao aplicativo.

Outra possibilidade relevante é a utilização do Flet, um framework que permite desenvolver aplicativos Flutter utilizando Python. Essa abordagem, descrita no portal da Hashtag Treinamentos, possibilita a criação de interfaces e aplicativos móveis por meio de scripts em Python, sem a necessidade de dominar a linguagem Dart. Embora o Flet seja uma alternativa prática para quem utiliza Python, ainda é uma ferramenta recente e limitada. Sua documentação é escassa, a comunidade pequena e os recursos mais simples, o que pode restringir o desenvolvimento de aplicações complexas (FLET, s.d.). Assim, o Flutter se apresenta como a opção mais completa e estável para futuras expansões do sistema.

2.7 Prototipação

Ao acessar o sistema, o usuário visualizará a tela de login, representada pela Figura 7. Nessa tela é necessário preencher as credenciais dos sistemas, sendo elas e-mail, senha e o tipo de usuário que está acessando.

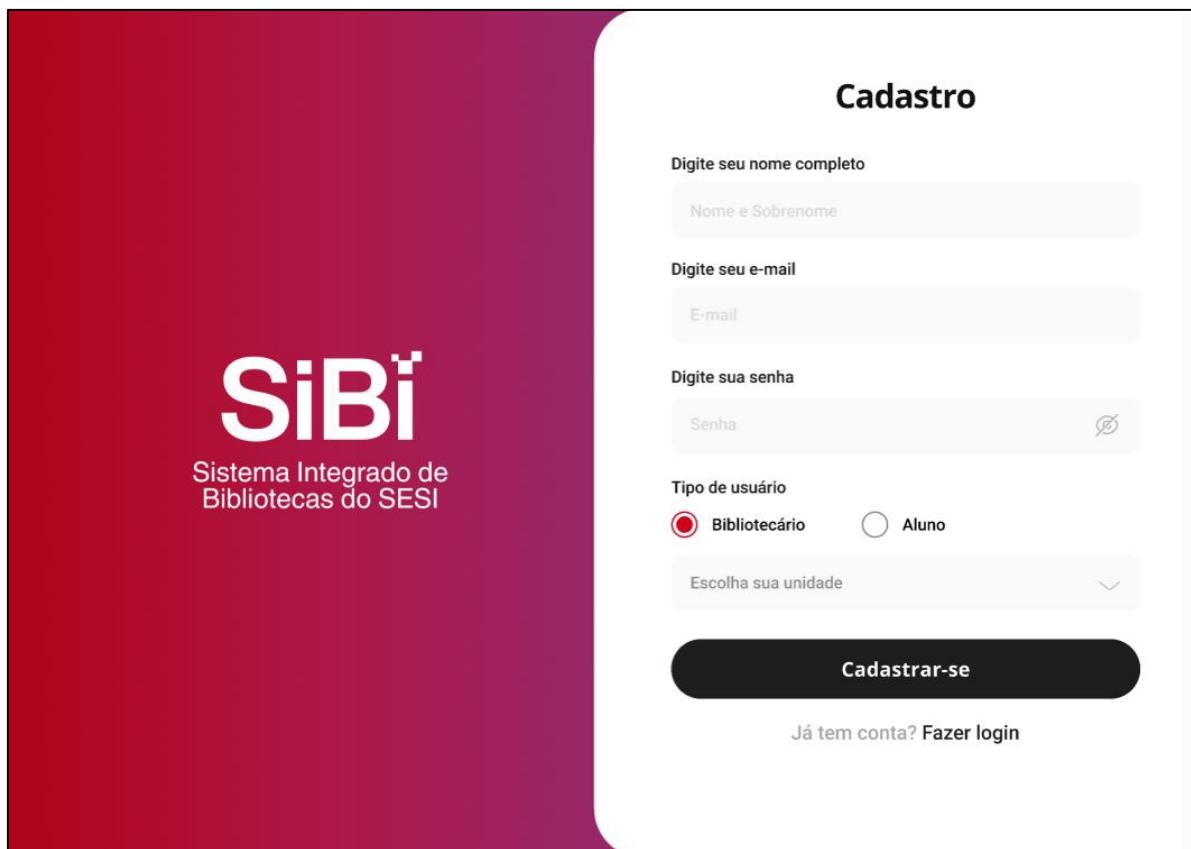
Figura 4 – Tela de Login



Fonte: Elaborado pelos autores (2025)

Caso as credenciais informadas não sejam válidas, o usuário será informado do erro, em caso de sucesso ocorrerá o redirecionamento para a home inicial. Caso o usuário não tenha conta na aplicação, será possível realizar o cadastro clicando em “Cadastre-se”, com isso ocorrerá o redirecionamento para a página de cadastro representada pela Figura 5.

Figura 5 – Tela de Cadastro

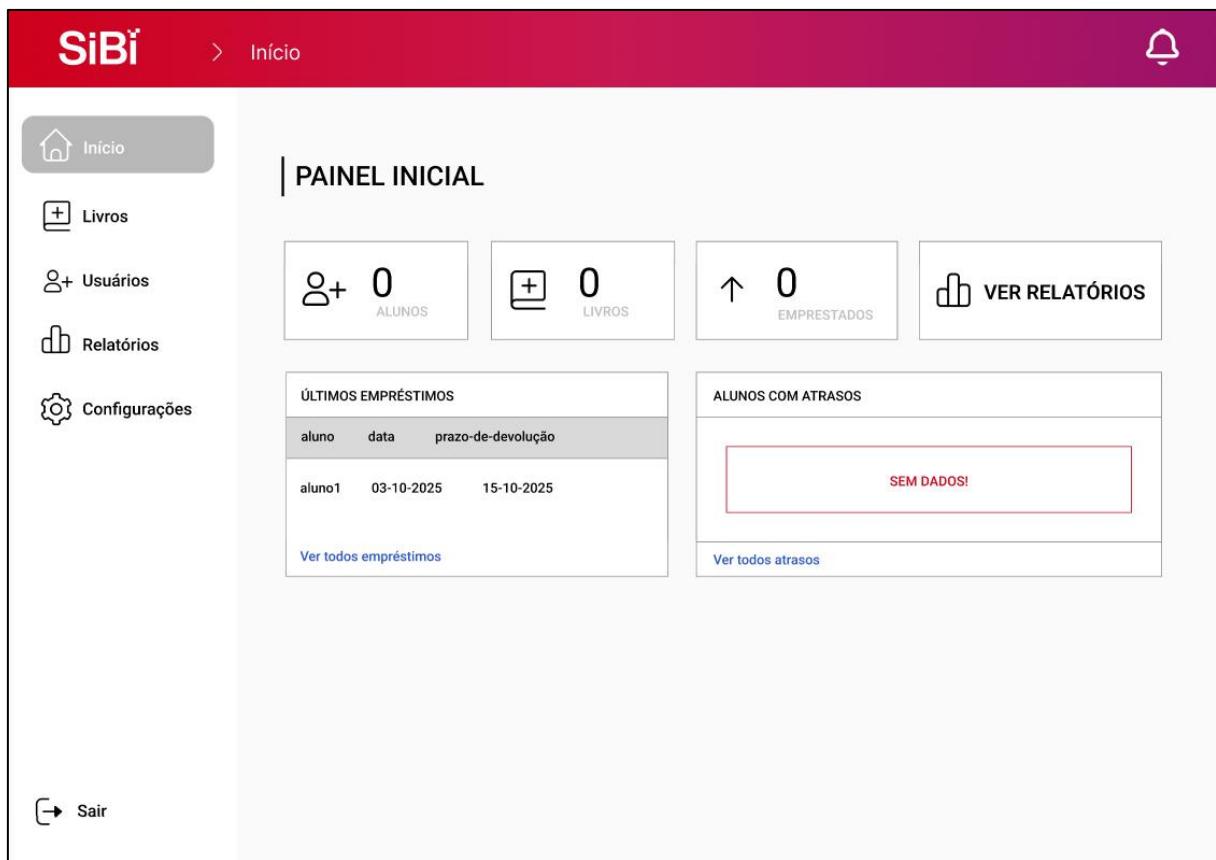


Fonte: Elaborado pelos autores (2025)

Acessando essa tela será necessário realizar o preenchimento de todos os campos disponíveis e clicar no botão “Cadastrar-se”. Caso todos os campos não estejam preenchidos será exibido uma modal informando quais campos faltam ser preenchidos, caso todos os campos estejam preenchidos será feita a verificação se já existe algum usuário cadastrado com o e-mail e nome completo digitado.

Se existir algum usuário já cadastrado será exibido uma modal com o aviso, caso contrário aparecerá uma mensagem informando que o usuário foi registrado com sucesso, além do redirecionamento para tela de login. Com o login realizado existem três cenários distintos para exibição da home inicial que variam de acordo com as permissões do usuário que podem variar entre “Aluno” e “Bibliotecário”.

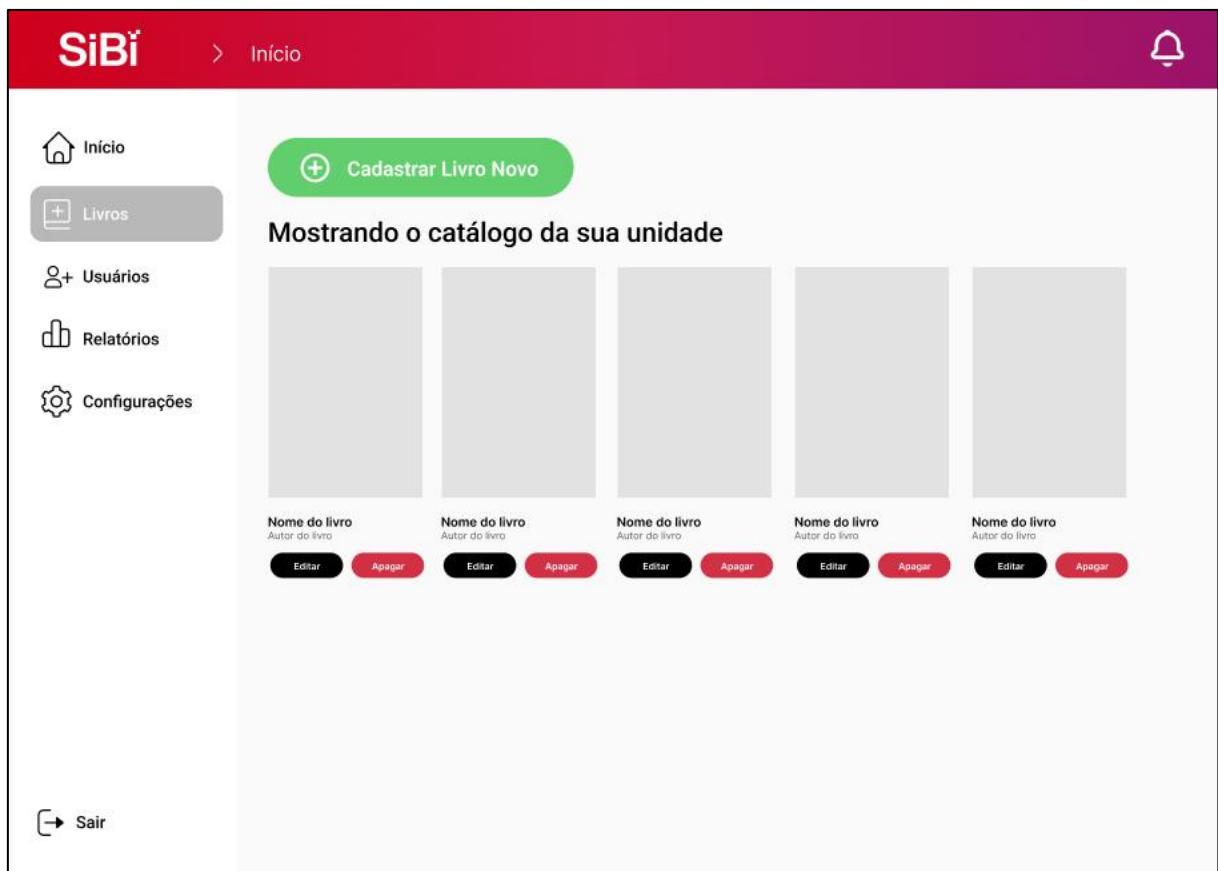
Figura 6 – Tela Inicial do Bibliotecário (Disponível somente para o bibliotecário)



Fonte: Elaborado pelos autores (2025)

A Figura 6 representa a visualização da tela principal do sistema SiBi no perfil de “Bibliotecário”. Essa tela exibe o menu lateral com os ícones de redirecionamento para a página de cadastro de livros, cadastro de alunos, visualização dos relatórios automáticos, configurações do perfil e o botão de saída da plataforma. É possível realizar a navegação pelo sistema de forma mais simplificada. A home inicial deverá conter cards com informações puxadas do banco de dados conectado.

Figura 7 – Página de Livros (Disponível somente para o bibliotecário)



Fonte: Elaborado pelos autores (2025)

A Figura 7 apresenta a página de cadastro de livros, que será utilizada exclusivamente pelo bibliotecário para realizar o gerenciamento do acervo. Nessa interface, o responsável poderá cadastrar novos livros, editar informações existentes e remover exemplares que não estejam mais disponíveis no sistema.

Os dados inseridos nesta página serão automaticamente integrados ao catálogo acessado pelos alunos, garantindo que todas as informações fiquem sempre atualizadas. Dessa forma, os estudantes poderão visualizar os títulos disponíveis e realizar reservas de forma rápida.

Figura 8 – Página de Usuários (Disponível somente para o bibliotecário)

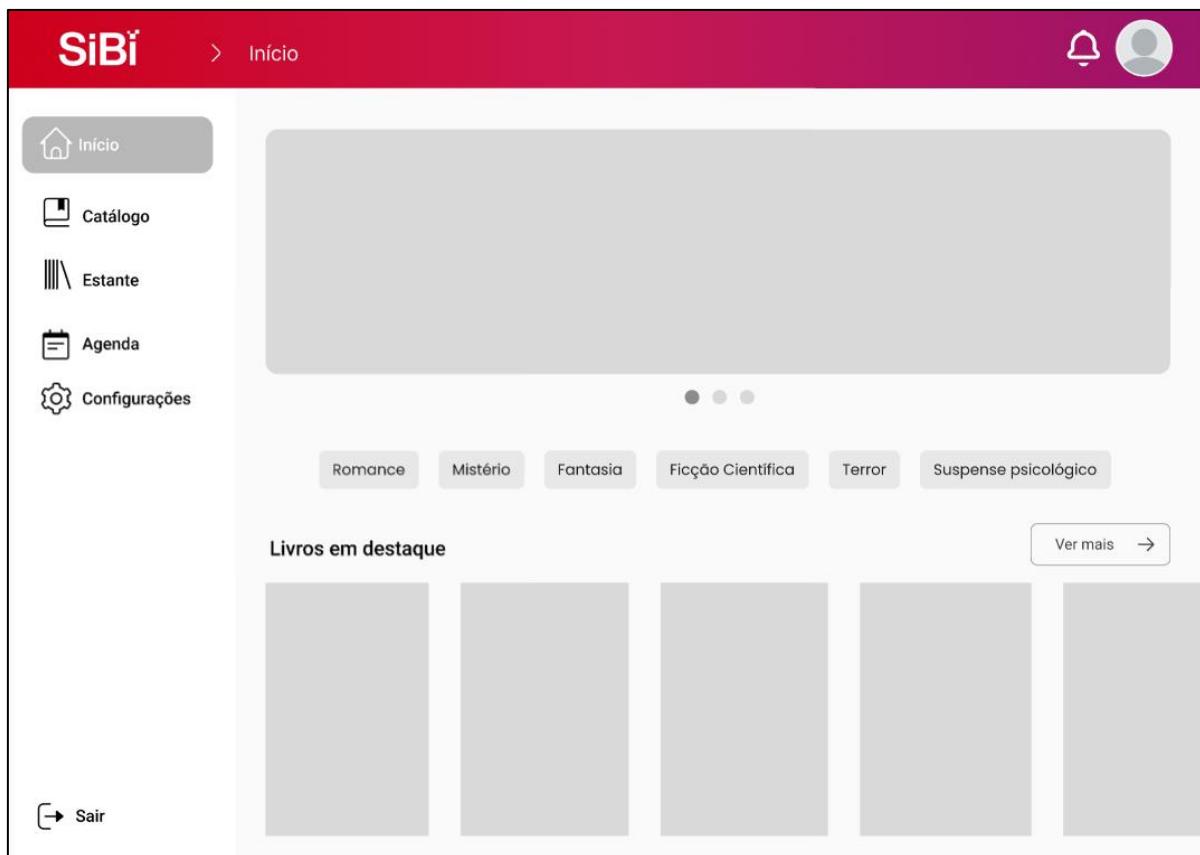
ID	Nome	E-mail	Senha	Unidade	Status
1	nome sobrenome	nome@gmail.com	\$12\$vtC6D976wD..	(CE 166) Escola..	ativo
2	nome sobrenome	nome@gmail.com	>j.{@5Sb(kvq>0%q7MzK	(CE 265) Escola..	inativo
3	nome sobrenome	nome@gmail.com	\$12\$vtC6D976wD..	(CE 094) Escola..	ativo
4	nome sobrenome	nome@gmail.com	\$12\$vtC6D976wD..	(CE 221) Escola..	ativo

Fonte: Elaborado pelos autores (2025)

A Figura 8 apresenta a página de gerenciamento de usuários, destinada ao bibliotecário para o controle do uso do sistema pelos alunos. Nessa área, é possível visualizar e administrar as reservas de livros, os empréstimos ativos, as devoluções pendentes e as confirmações de devolução já realizadas. Além disso, o bibliotecário pode consultar o histórico de empréstimos e o status individual de cada aluno.

Futuramente, esta seção também permitirá a atualização de dados dos usuários e o acompanhamento de pendências, ampliando o controle e a eficiência na gestão da biblioteca.

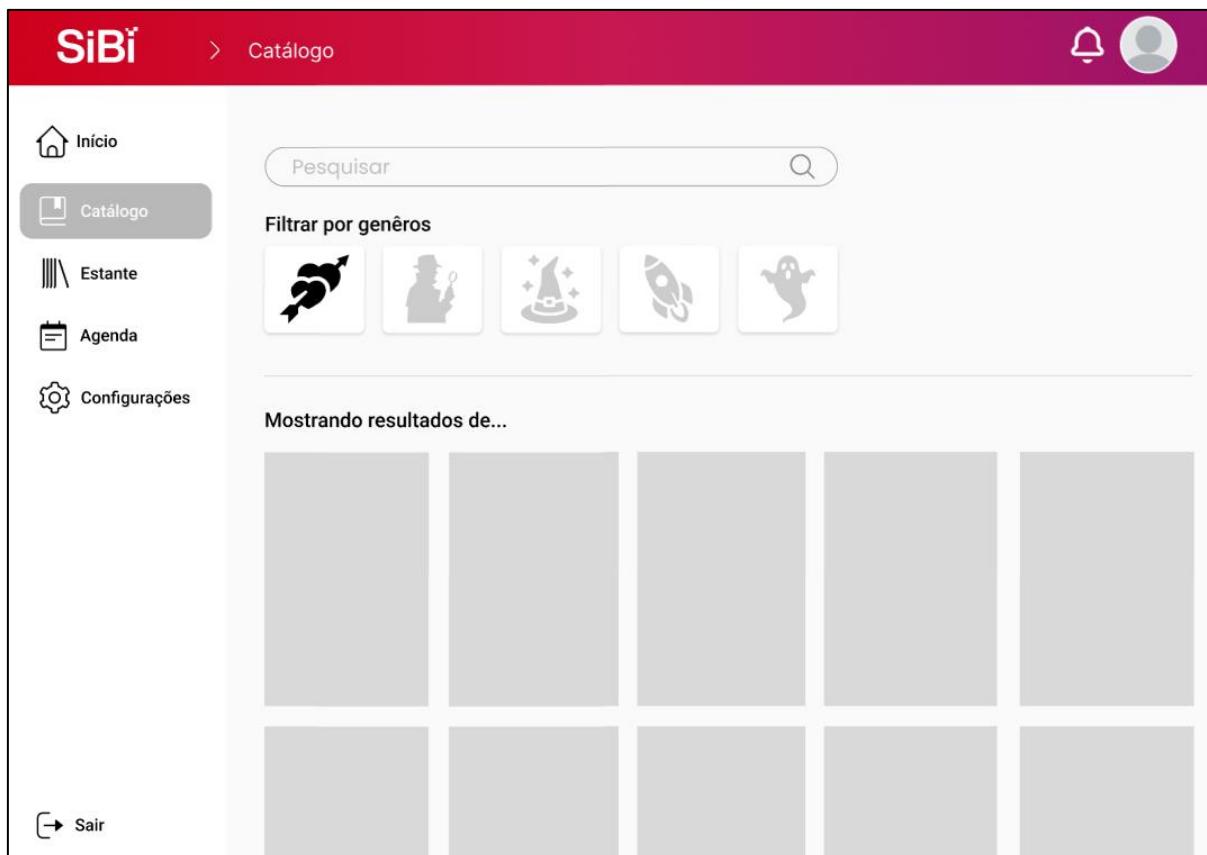
Figura 9 – Tela Inicial do Aluno (Disponível somente para o aluno)



Fonte: Elaborado pelos autores (2025)

A Figura 9 apresenta a tela inicial do aluno, que exibe um carrossel com propagandas de eventos, novidades e avisos da biblioteca. Além disso, o aluno pode visualizar livros em destaque e acessar o catálogo completo para explorar ou reservar exemplares.

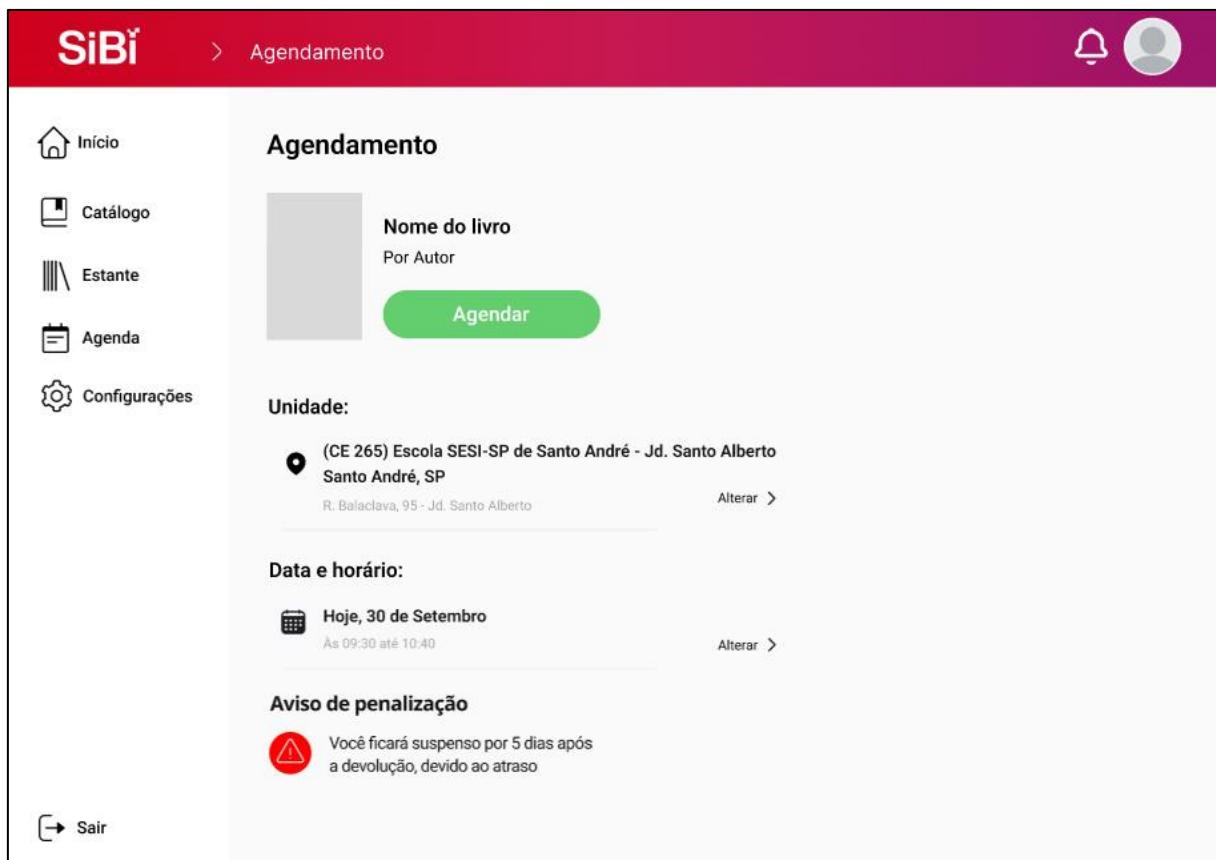
Figura 10 – Catálogo de Livros (Disponível somente para o aluno)



Fonte: Elaborado pelos autores (2025)

A Figura 10 mostra o catálogo de livros, página utilizada pelos alunos para visualizar os exemplares disponíveis na biblioteca. Nela, é possível ver detalhes dos livros — como nome, autor, sinopse e capa — além de solicitar empréstimos, renovar e devolver obras. Futuramente, essa área será interativa, permitindo que os alunos gerenciem seus empréstimos diretamente pelo catálogo.

Figura 11 – Agendamento (Disponível somente para o aluno)



Fonte: Elaborado pelos autores (2025)

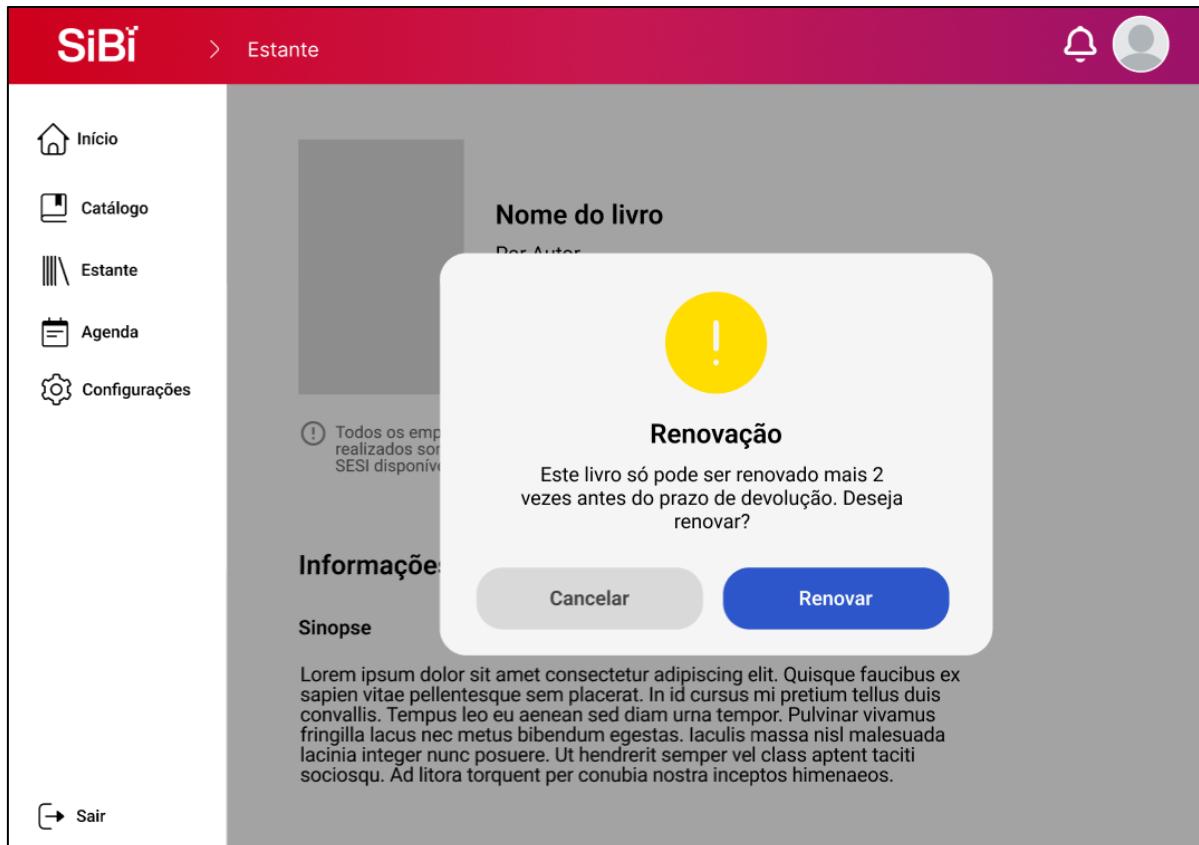
A Figura 11 representa a visualização da tela de detalhamento de um livro no sistema SiBi. Essa tela é exibida quando o usuário clica sobre um título listado no catálogo, permitindo visualizar informações completas sobre a obra, como nome, autor, descrição, disponibilidade e unidade em que o exemplar se encontra.

Caso o livro esteja disponível para empréstimo, o sistema apresentará o botão “Fazer empréstimo”, permitindo ao aluno realizar o agendamento do empréstimo. Já se o exemplar estiver vinculado ao usuário, a tela exibirá a opção “Fazer devolução”, direcionando-o para o processo de agendamento da devolução.

Ao selecionar qualquer uma dessas opções, o aluno é encaminhado para a tela de Agendamento, onde poderá escolher a unidade, data e horário para concluir a solicitação. As confirmações de agendamento realizadas pelos alunos serão enviadas automaticamente ao bibliotecário da unidade correspondente ao livro, garantindo o acompanhamento e controle adequado das operações.

Essa funcionalidade proporciona maior praticidade, organização e segurança no gerenciamento dos empréstimos e devoluções dentro do sistema.

Figura 12 – Estante (Disponível somente para o aluno)



Fonte: Elaborado pelos autores (2025)

A Figura 12 apresenta a página “Minha Estante”, onde os alunos podem gerenciar os livros emprestados. Nessa área, é possível visualizar os empréstimos ativos, renovar livros de forma prática e realizar devoluções por meio da aba de agendamento. Além disso, o sistema indica se há livros atrasados ou pendentes.

Em breve, essa página será mais interativa, facilitando o acompanhamento e controle dos empréstimos e devoluções.

2.8 Resultados Obtidos

Neste tópico, será apresentado os resultados obtidos ao longo da fase de desenvolvimento do projeto, destacando as principais funções e suas respectivas telas.

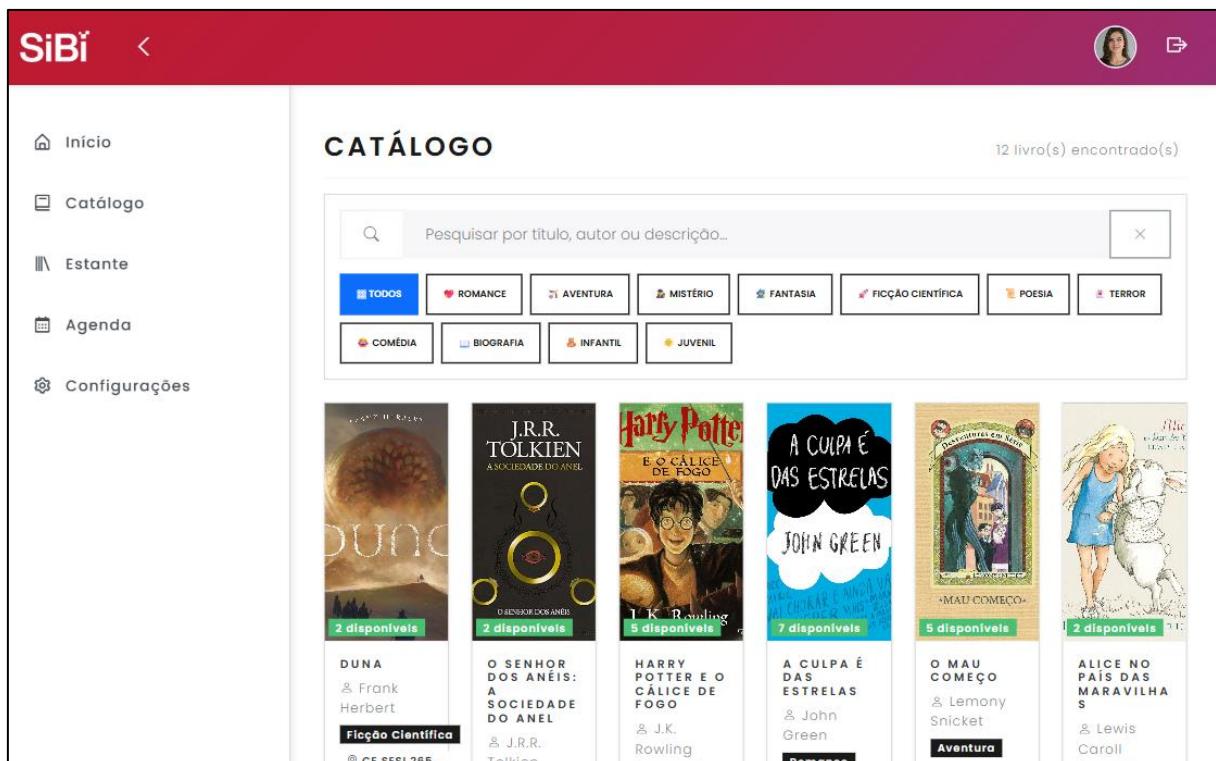
Figura 13 – Página Inicial (Aluno)



Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a tela inicial do aluno do nosso projeto, Sistema Integrado de Bibliotecas (SiBi), nela, o estudante visualiza um banner principal com divulgação de eventos e campanhas institucionais, funcionando como um espaço dinâmico para anúncios importantes da biblioteca. Logo abaixo, são exibidos os livros disponíveis na unidade, com capas destacadas para facilitar o reconhecimento e incentivar a exploração do acervo. O aluno também conta com um botão de acesso rápido ao catálogo completo, permitindo consultar mais obras, obter detalhes e realizar reservas de forma prática e intuitiva.

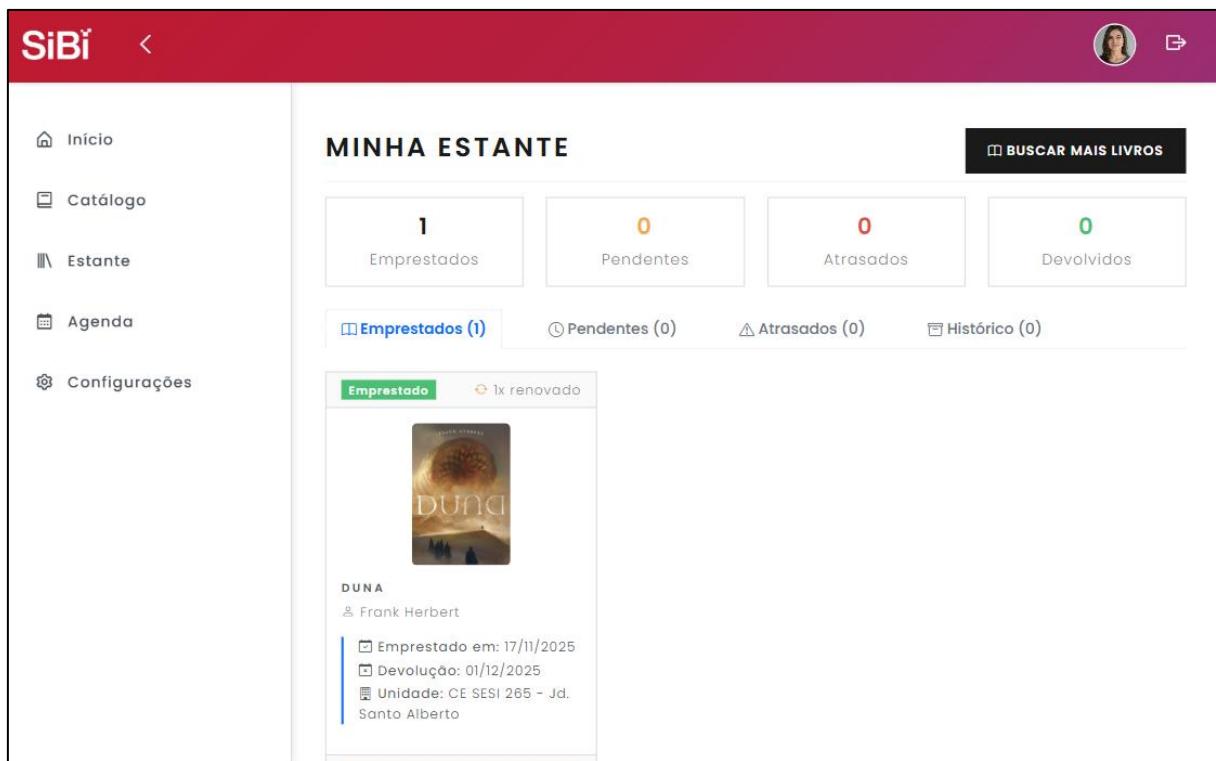
Figura 14 – Catálogo de Livros (Aluno)



Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página de Catálogo do Sistema Integrado de Bibliotecas (SiBi), onde o aluno pode visualizar os livros disponíveis na unidade de forma organizada e acessível. A interface conta com uma barra de pesquisa para localizar obras por título, autor ou descrição, além de filtros por gênero literário, como aventura, romance, mistério, fantasia e outros. Os livros são exibidos com suas respectivas capas, títulos, autores e quantidade de exemplares disponíveis, facilitando a identificação das obras. Essa página permite ao aluno explorar o acervo de maneira prática e, futuramente, oferecerá recursos interativos para solicitação, renovação e gerenciamento de empréstimos diretamente pelo catálogo.

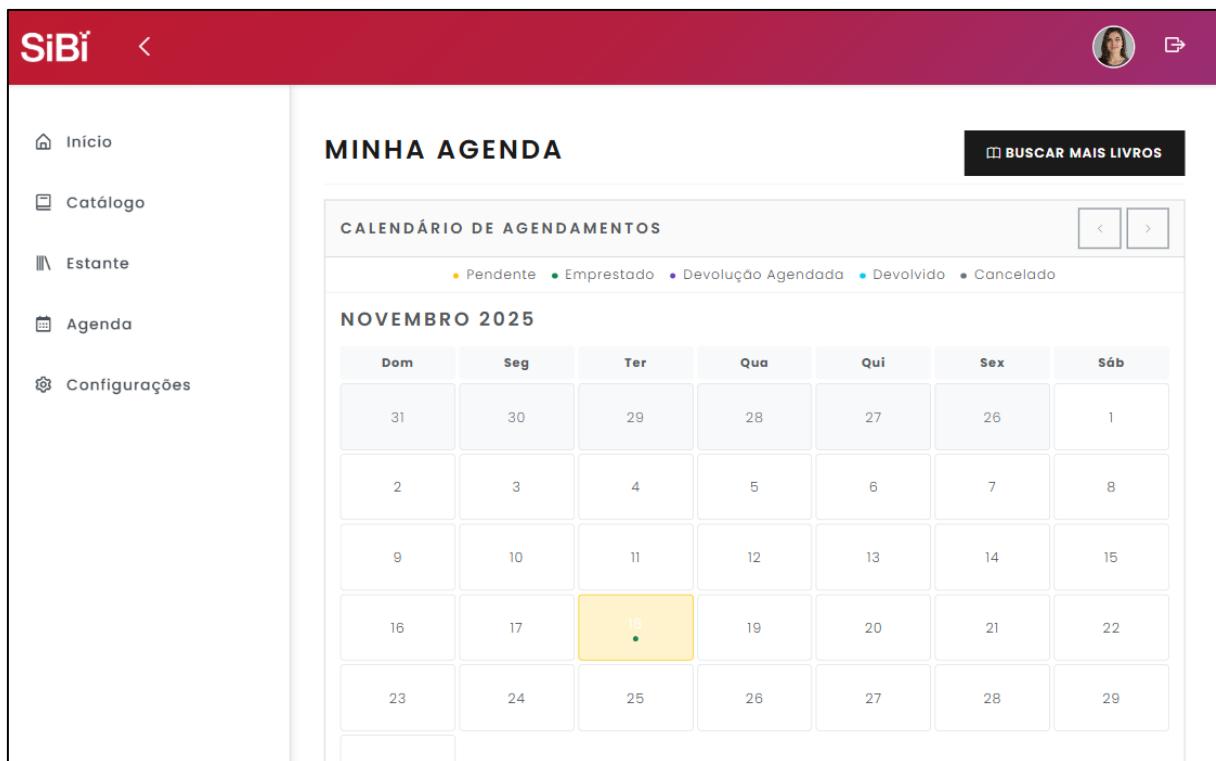
Figura 15 – Estante (Aluno)



Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página “Minha Estante” do Sistema Integrado de Bibliotecas (SiBi), onde o aluno pode visualizar e gerenciar seus empréstimos de maneira clara e organizada. No topo, são exibidos indicadores de livros emprestados, pendentes, atrasados e devolvidos, oferecendo uma visão geral imediata da situação do usuário. Na aba de “Emprestados”, o aluno tem acesso aos detalhes da obra ativa, incluindo capa, título, autor, data de empréstimo, data de devolução e unidade da biblioteca, além da informação de que o livro já foi renovado. A interface também conta com um botão para buscar mais livros, permitindo que o estudante acesse rapidamente o catálogo.

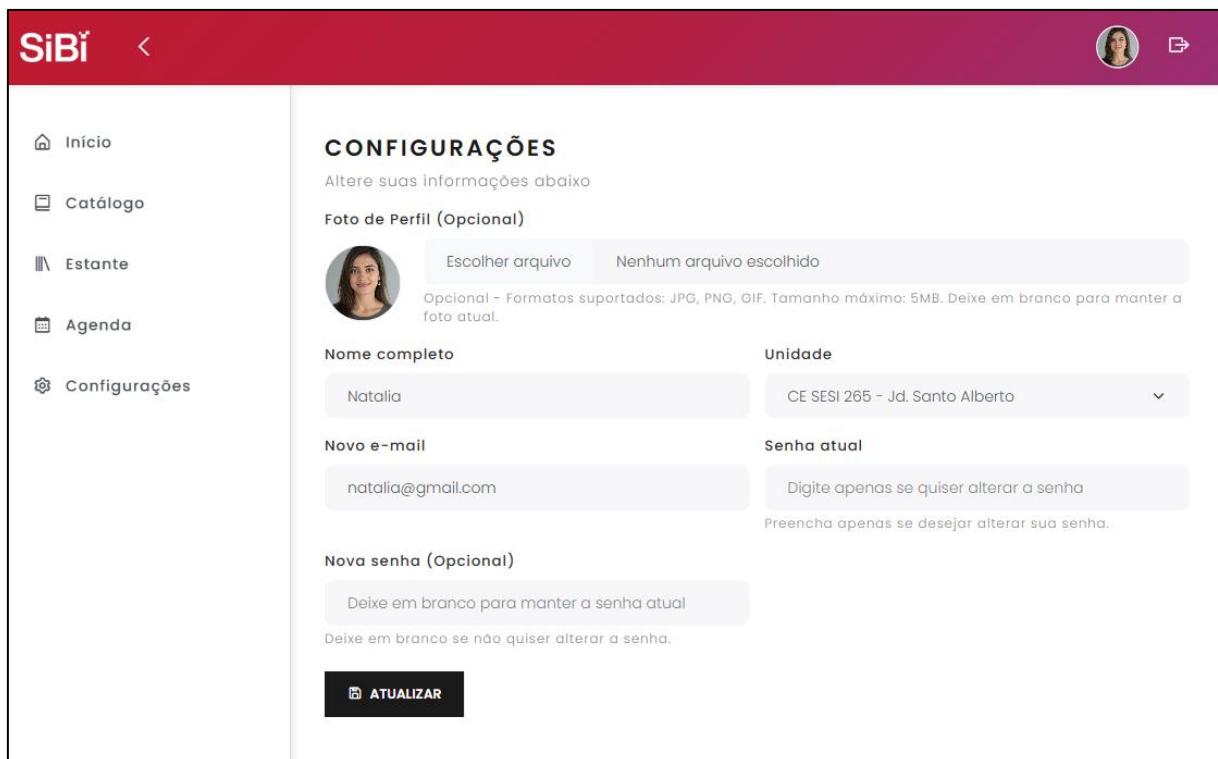
Figura 16 – Agenda (Aluno)



Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página Agenda do aluno no Sistema Integrado de Bibliotecas (SIBi), onde é exibido o calendário completo de agendamentos. Nessa tela, o estudante pode visualizar de forma organizada todos os seus empréstimos, devoluções e reservas, identificados por cores que indicam o status de cada agendamento, como pendente, devolvido, cancelado ou devolução programada. O calendário permite acompanhar facilmente os compromissos do mês, destacando os dias que possuem atividades relacionadas à biblioteca. Além disso, a página oferece navegação entre meses e um botão para buscar novos livros, facilitando o planejamento e o controle das leituras pelo aluno.

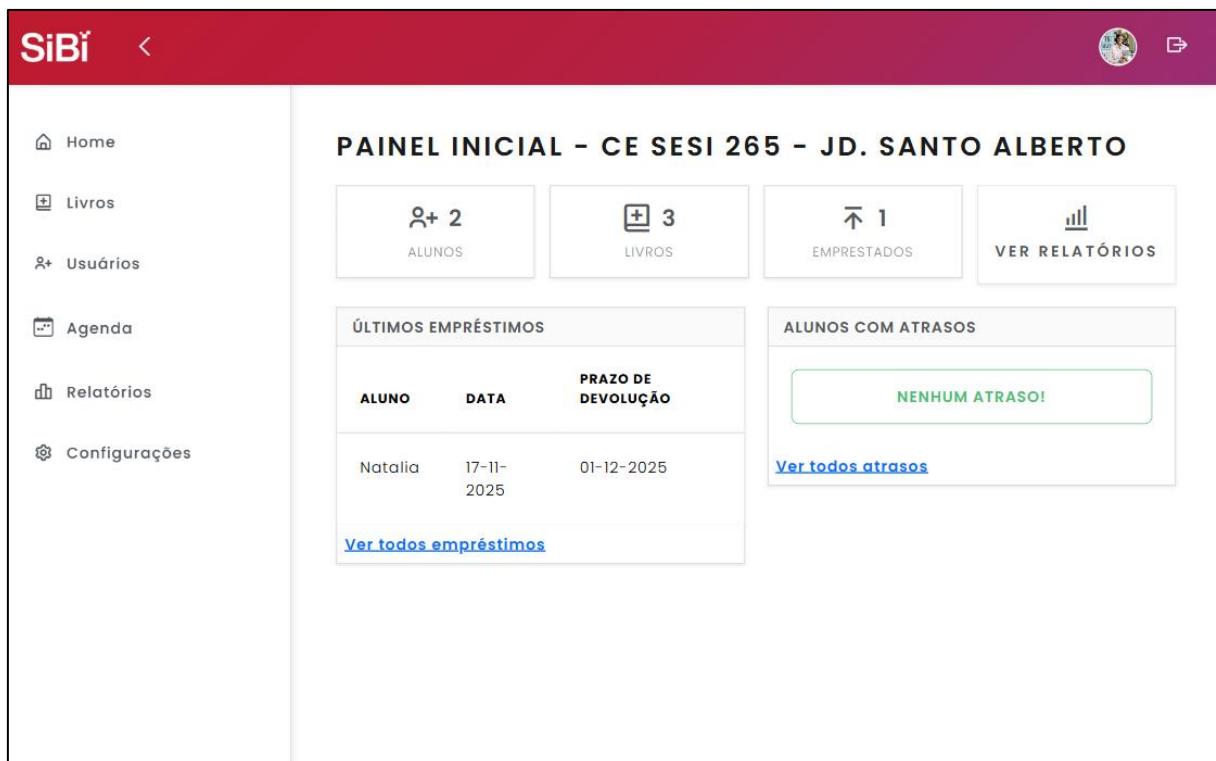
Figura 17 – Configurações (Aluno)



Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página Configurações do aluno dentro do Sistema Integrado de Bibliotecas (SIBi), onde é possível atualizar informações pessoais e ajustar dados da conta. Nessa tela, o usuário pode alterar sua foto de perfil, atualizar o nome completo, modificar o e-mail cadastrado e selecionar sua unidade escolar. Caso deseje, também pode trocar a senha, inserindo a senha atual e definindo uma nova. A interface organiza os campos de forma clara, permitindo que o aluno gerencie seus dados com facilidade e salve todas as alterações por meio do botão de atualização.

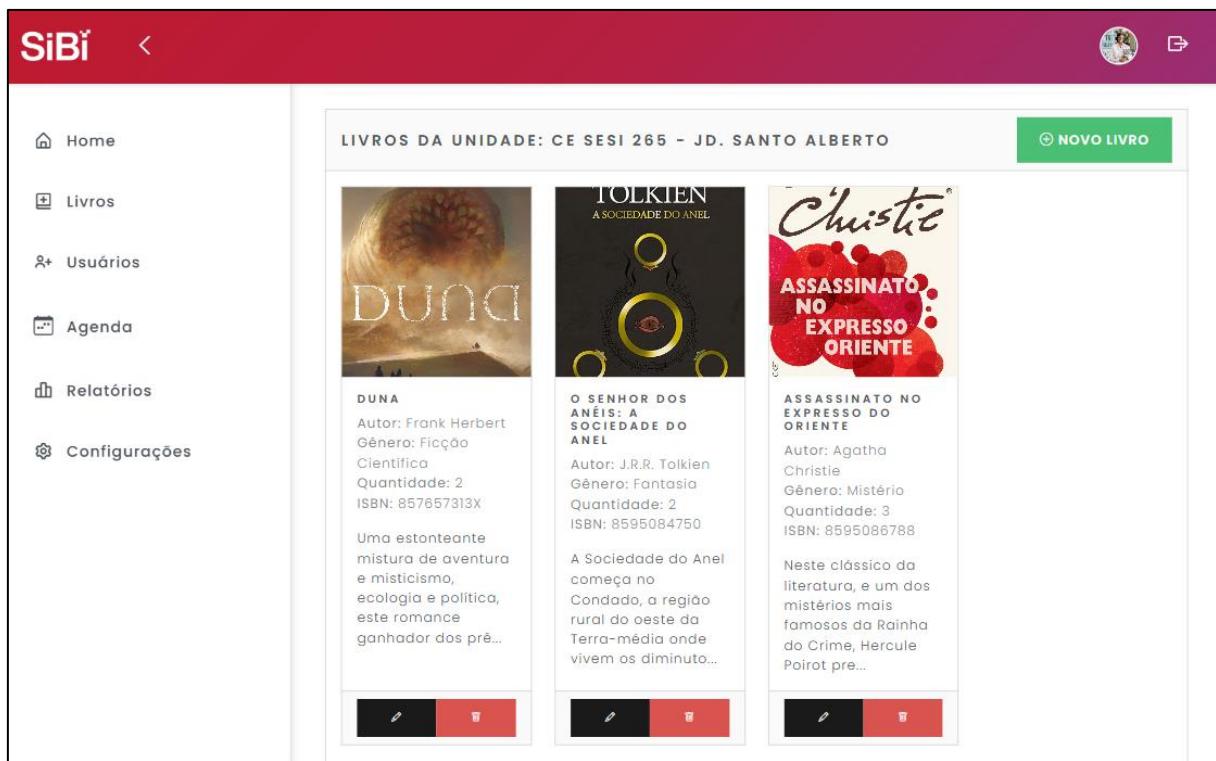
Figura 18 – Painel Inicial (Bibliotecário)



Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página inicial do bibliotecário no Sistema Integrado de Bibliotecas (SIBi), onde são exibidos indicadores gerais da unidade, como o número total de alunos cadastrados, quantidade de livros disponíveis e exemplares atualmente emprestados. A tela também oferece acesso rápido aos relatórios administrativos. Logo abaixo, o bibliotecário visualiza uma tabela com os últimos empréstimos realizados, incluindo nome do aluno, data do empréstimo e prazo de devolução. Ao lado, há um painel que informa se existem alunos com devoluções atrasadas, facilitando o acompanhamento e a gestão do acervo. A interface organiza as informações de forma clara e objetiva, auxiliando o bibliotecário no controle diário da biblioteca.

Figura 19 – Livros (Bibliotecário)



Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página Livros do bibliotecário no Sistema Integrado de Bibliotecas (SIBi), onde são exibidos todos os exemplares cadastrados na unidade. Cada livro aparece organizado em cartões contendo capa, título, autor, gênero, quantidade disponível e ISBN, além de uma breve descrição da obra. A interface oferece botões de edição e exclusão, permitindo ao bibliotecário gerenciar o acervo de forma rápida e prática. No topo da página, há também o botão “Novo Livro”, que possibilita adicionar novos títulos ao sistema. Essa visualização clara e segmentada facilita a administração do catálogo e o acompanhamento das obras disponíveis na biblioteca.

Figura 20 – Gerenciar Usuários (Bibliotecário)

ID	NOME	EMAIL	TIPO DE USUÁRIO	UNIDADE	AÇÕES
1	Natalia	natalia@gmail.com	Aluno	CE SESI 265 - Jd. Santo Alberto	<button>EDITAR</button> <button>EXCLUIR</button>
2	Vivian	vivian@gmail.com	Bibliotecário	CE SESI 265 - Jd. Santo Alberto	<button>EDITAR</button> <button>EXCLUIR</button>
3	Bruna	bruna@gmail.com	Aluno	CE SESI 265 - Jd. Santo Alberto	<button>EDITAR</button> <button>EXCLUIR</button>

Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página Gerenciamento de Usuários do bibliotecário no Sistema Integrado de Bibliotecas (SIBi). Nessa tela, o responsável pela unidade pode visualizar todos os usuários cadastrados, organizados em uma tabela que exibe ID, nome, e-mail, tipo de usuário e unidade correspondente. A interface diferencia visualmente alunos e bibliotecários por meio de etiquetas coloridas, facilitando a identificação rápida dos perfis. Ao lado de cada registro, há botões de edição e exclusão, permitindo ajustar informações ou remover usuários quando necessário. No topo da página, o botão “Novo Usuário” possibilita adicionar novos cadastros ao sistema, tornando o gerenciamento simples, direto e eficiente.

Figura 21 – Agenda de Empréstimos e Devoluções (Bibliotecário)

The screenshot shows the SIBi system's library management interface. On the left, a sidebar lists navigation options: Home, Livros, Usuários, Agenda, Relatórios, and Configurações. The main content area is titled "AGENDA DE EMPRÉSTIMOS E DEVOLUÇÕES". It features a summary bar with counts for Pending (0), Borrowed (1), Scheduled Return (0), Returned (0), Canceled (0), and Total (1). Below this are filtering and sorting tools: "Filtrar por Status" (Todos os Status), "Ordenar por" (Data (Mais Próxima)), and a search bar ("Buscar: Aluno ou livro..."). A detailed view of a single loan record is shown, including the status "Emprestado", the item "DUNA", author "Frank Herbert", category "Ficção Científica", borrower "Aluno: Natalia", email "natalia@gmail.com", and scheduled return information ("Agendado para: 18/11/2025" and "Horário: Manhã (9:30-10:40)").

Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página Agenda de Empréstimos e Devoluções do bibliotecário no Sistema Integrado de Bibliotecas (SIBi). Nessa tela, o responsável pela unidade tem acesso a um painel com indicadores que mostram, de forma organizada, a quantidade de empréstimos pendentes, em andamento, devoluções agendadas, devoluções concluídas, registros cancelados e o total de movimentações.

Logo abaixo, a interface oferece opções de filtro por status, ordenar por data e busca por aluno ou livro, facilitando o gerenciamento das atividades. Cada registro exibido na agenda apresenta informações detalhadas sobre o empréstimo, como o status, título da obra, autor, categoria e dados do aluno, além da data e horário agendados. Essa página centraliza o acompanhamento de todas as movimentações da biblioteca, tornando o controle diário mais prático, visual e eficiente.

Figura 22 – Relatórios (Bibliotecário)

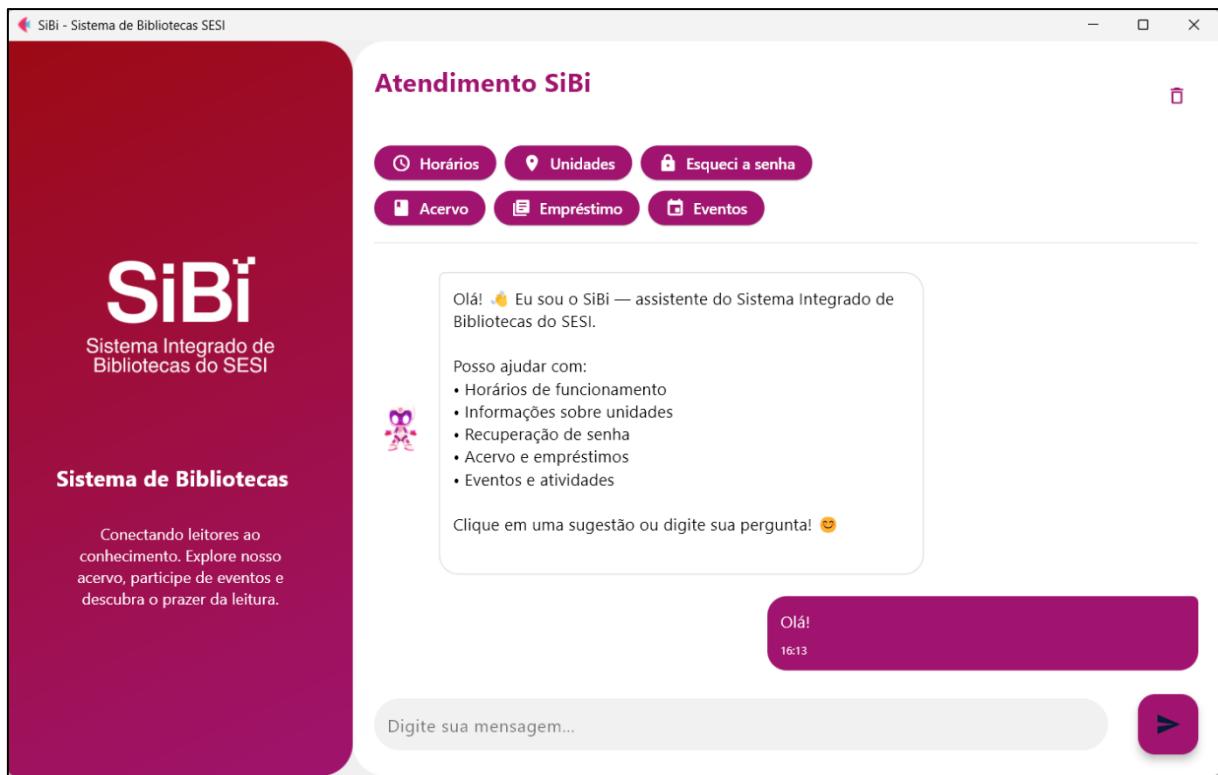


Fonte: Desenvolvido pelos autores (2025)

A imagem apresenta a página Relatórios do bibliotecário no Sistema Integrado de Bibliotecas (SIBi). Nessa tela, o responsável pela unidade pode gerar análises completas sobre o uso da biblioteca, visualizando gráficos e tabelas que organizam os dados de empréstimos e devoluções dos últimos 15 dias. No topo da interface, os botões “Gerar Relatório” e “Exportar em PDF” permitem ao bibliotecário criar e salvar documentos detalhados para acompanhamento interno ou apresentações.

A página exibe gráficos que mostram o volume recente de empréstimos e devoluções, seguidos por tabelas que listam os detalhes de cada movimentação por data. Além disso, são apresentados relatórios complementares, como o Top 5 livros mais emprestados e o gráfico de livros mais emprestados por gênero, oferecendo uma visão estratégica sobre o comportamento de leitura dos usuários. Essa estrutura facilita o monitoramento das tendências da biblioteca e apoia a tomada de decisões, tornando a gestão mais organizada, visual e eficiente.

Figura 23 – Interface do Chat de Atendimento com respostas automáticas do SIBi



Fonte: Desenvolvido pelos autores (2025)

A imagem mostra a tela do chat de atendimento do Sistema Integrado de Bibliotecas (SiBi), desenvolvido utilizando o Flet. Nela é possível visualizar a interface do assistente virtual (usando a imagem do mascote do SiBi, chamado “Booki”), que oferece respostas rápidas sobre horários de funcionamento, informações das unidades, recuperação de senha, acervo, empréstimos e divulgação de eventos.

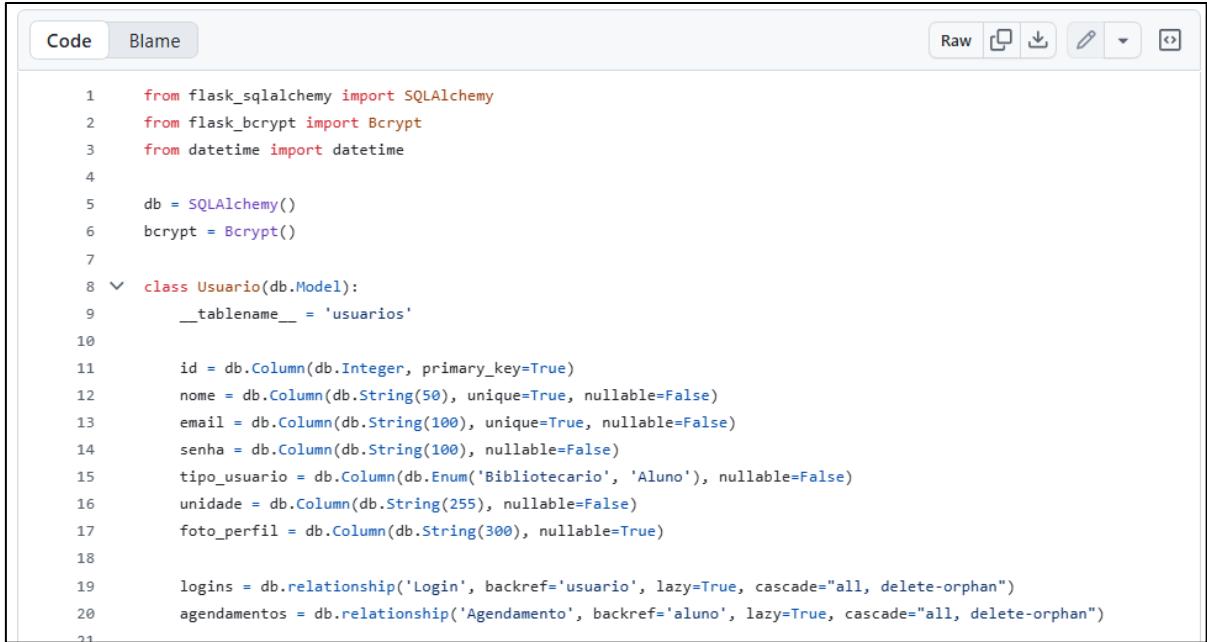
Na parte superior, aparecem os botões de acesso rápido para cada uma dessas funções, como “Horários”, “Unidades”, “Esqueci minha senha”, “Acervo”, “Empréstimos” e “Eventos”. Ao centro, o ChatBot apresenta uma mensagem inicial explicando os tipos de atendimento disponíveis, organizada em formato de balão de conversa.

2.8.1 Codificação

O Sistema Integrado de Bibliotecas (SIBI) foi desenvolvido em Python, utilizando Flask, framework web principal, com o SQLAlchemy para o mapeamento de objeto-

relacional e o MySQL para o banco de dados. A aplicação segue uma arquitetura modular baseada em Blueprints do Flask, permitindo a separação de funcionalidades em módulos independentes.

Figura 24 – Estrutura e Modelagem de Dados



The screenshot shows a GitHub code editor interface with a tab bar at the top labeled 'Code' and 'Blame'. On the right side of the header are buttons for 'Raw', copy, cut, paste, edit, and refresh. The main area contains the following Python code:

```
1  from flask_sqlalchemy import SQLAlchemy
2  from flask_bcrypt import Bcrypt
3  from datetime import datetime
4
5  db = SQLAlchemy()
6  bcrypt = Bcrypt()
7
8  class Usuario(db.Model):
9      __tablename__ = 'usuarios'
10
11     id = db.Column(db.Integer, primary_key=True)
12     nome = db.Column(db.String(50), unique=True, nullable=False)
13     email = db.Column(db.String(100), unique=True, nullable=False)
14     senha = db.Column(db.String(100), nullable=False)
15     tipo_usuario = db.Column(db.Enum('Bibliotecario', 'Aluno'), nullable=False)
16     unidade = db.Column(db.String(255), nullable=False)
17     foto_perfil = db.Column(db.String(300), nullable=True)
18
19     logins = db.relationship('Login', backref='usuario', lazy=True, cascade="all, delete-orphan")
20     agendamentos = db.relationship('Agendamento', backref='aluno', lazy=True, cascade="all, delete-orphan")
21
```

Fonte: Desenvolvido pelos autores (2025) - GitHub

Arquivo fundamental, pois ele define a estrutura do seu banco de dados e as relações entre as entidades. Mostra como ocorre a Modelagem de Dados dos requisitos do sistema de biblioteca em um modelo de dados relacional (SQLAlchemy) e seu controle de fluxo.

Figura 25 – Sistema de Autenticação e Login

```

71     # ===== LOGIN =====
72     @app.route('/login', methods=['GET', 'POST'])
73     def login():
74         if request.method == 'POST':
75             email = request.form['email']
76             senha = request.form['senha']
77             tipo_usuario = request.form['tipo_usuario']
78
79             user = Usuario.query.filter_by(email=email, tipo_usuario=tipo_usuario).first()
80
81             if user and bcrypt.check_password_hash(user.senha, senha):
82                 session['logged_in'] = True
83                 session['usuario_id'] = user.id
84                 session['nome_completo'] = user.nome
85                 session['nome'] = user.nome
86                 session['tipo_usuario'] = user.tipo_usuario
87                 session['unidade'] = user.unidade
88                 session['email'] = user.email
89                 session['foto_perfil'] = user.foto_perfil
90
91                 login_registro = Login(usuario_id=user.id)
92                 db.session.add(login_registro)
93                 db.session.commit()
94

```

Fonte: Desenvolvido pelos autores (2025) – GitHub

Esta função gerencia o processo de autenticação. Ela verifica se o usuário existe no banco de dados e se a senha fornecida corresponde ao hash armazenado. Em caso de sucesso, cria uma sessão com os dados do usuário, permitindo o acesso às funcionalidades do sistema.

Figura 26 – Sistema de Autenticação e Login

```

73     def login():
74
75         # Em vez de redirect, renderiza a splash diretamente
76         destino = 'home_bibliotecario' if user.tipo_usuario == 'Bibliotecario' else 'home_aluno'
77         destino_url = url_for(destino)
78         return render_template('splash.html', destino_url=destino_url)
79     else:
80         flash('Usuário ou senha inválidos', 'error')
81         return render_template('login.html')
82
83     return render_template('login.html')
84

```

Fonte: Desenvolvido pelos autores (2025) – GitHub

Já esse trecho de código protege as rotas restritas, garantindo que apenas usuários autenticados com o papel adequado (Bibliotecário ou Aluno) possam acessar

determinadas funcionalidades. O sistema implementa uma separação clara de permissões entre os dois tipos de usuários.

Figura 27 – Sistema de Cadastro de Usuários



The screenshot shows a GitHub code editor interface with the 'Code' tab selected. The code is written in Python and defines a function for user registration. It includes logic to handle both GET and POST requests, validate email uniqueness, and hash passwords using bcrypt. The code is annotated with line numbers from 105 to 127.

```
105     # ===== CADASTRO =====
106     @app.route('/cadastro', methods=['GET', 'POST'])
107     def cadastro():
108         unidades = [
109             "CE SESI 265 - Jd. Santo Alberto",
110             "CE SESI 094 - Vila Clarice",
111             "CE SESI 166 - Santa Terezinha",
112             "CE SESI 221 - Parque Jaçatuba"
113         ]
114
115         if request.method == 'POST':
116             nome = request.form['nome']
117             email = request.form['email']
118             senha = request.form['senha']
119             tipo_usuario = request.form['tipo_usuario']
120             unidade = request.form['unidade']
121
122             usuario_existente = Usuario.query.filter_by(email=email).first()
123             if usuario_existente:
124                 flash('Este email já está cadastrado!', 'error')
125                 return render_template('cadastro.html', unidades=unidades)
126
127             senha_hash = bcrypt.generate_password_hash(senha).decode('utf-8')
```

Fonte: Desenvolvido pelos autores (2025) – GitHub

Esta função gerencia o processo de cadastro de novos usuários no sistema. Ela define as unidades disponíveis para seleção e lida com tantas requisições GET (exibição do formulário) quanto POST (processamento dos dados).

O sistema verifica se o email já está cadastrado no banco de dados, prevenindo duplicatas e mantendo a unicidade das contas. Caso o email exista, retorna uma mensagem de erro ao usuário. As senhas são criptografadas usando bcrypt antes do armazenamento, garantindo a segurança das credenciais dos usuários mesmo em caso de violação do banco de dados.

Figura 28 – Sistema do Bibliotecário

The screenshot shows a GitHub code editor interface with a dark theme. At the top, there are tabs for 'Code' and 'Blame'. On the right side, there are several icons for file operations: Raw, Copy, Download, Edit, and others. The main area contains the following Python-like template code:

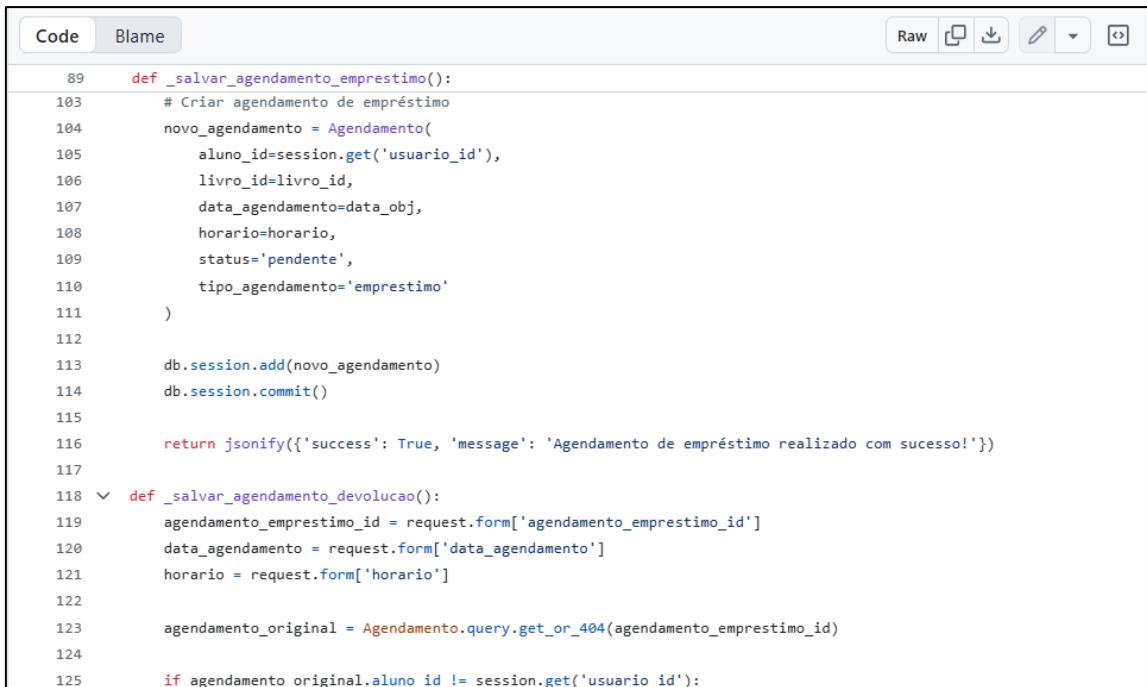
```
{% for usuario in usuarios %}
<tr>
    <td>{{ usuario.id }}</td>
    <td>{{ usuario.nome }}</td>
    <td>{{ usuario.email }}</td>
    <td>
        <span class="badge" {% if usuario.tipo_usuario == 'Bibliotecario' %}bg-warning{% else %}bg-info{% endif %}>
            {{ usuario.tipo_usuario }}
        </span>
    </td>
    <td>{{ usuario.unidade }}</td>
    <td>
        <a href="{{ url_for('gerenciar_usuarios.editar_usuario', id=usuario.id) }}">
            class="btn btn-primary btn-sm">
                <i class="bi bi-pencil"></i> Editar
        </a>
        <a href="{{ url_for('gerenciar_usuarios.excluir_usuario', id=usuario.id) }}">
            class="btn btn-danger btn-sm"
            onclick="return confirm('Tem certeza que deseja excluir este usuário?')"
            <i class="bi bi-trash"></i> Excluir
        </a>
    </td>
</tr>
{% endfor %}
```

Fonte: Desenvolvido pelos autores (2025) – GitHub

Esta função gerencia a listagem de usuários, garantindo que o bibliotecário visualize apenas os usuários da sua própria unidade. O bibliotecário poderá cadastrar um novo usuário, editar e excluir os usuários existentes.

O arquivo gerenciar_livros.html segue a mesma estrutura e possui as mesmas funções do arquivo explicado acima.

Figura 29 – Sistema de Agendamentos



The screenshot shows a GitHub code editor interface with the 'Code' tab selected. The code is written in Python and defines two functions: '_salvar_agendamento_emprestimo()' and '_salvar_agendamento_devolucao()'. The first function creates a new appointment (Agendamento) with parameters like aluno_id, livro_id, data_agendamento, horario, status ('pendente'), and tipo_agendamento ('emprestimo'). It then adds the appointment to a database session and commits the changes. The second function retrieves an appointment by its ID from the database and updates its status and due date based on the form data. It also checks if the appointment's student ID matches the user's ID.

```
89     def _salvar_agendamento_emprestimo():
103         # Criar agendamento de empréstimo
104         novo_agendamento = Agendamento(
105             aluno_id=session.get('usuario_id'),
106             livro_id=livro_id,
107             data_agendamento=data_obj,
108             horario=horario,
109             status='pendente',
110             tipo_agendamento='emprestimo'
111         )
112
113         db.session.add(novo_agendamento)
114         db.session.commit()
115
116         return jsonify({'success': True, 'message': 'Agendamento de empréstimo realizado com sucesso!'})
117
118     def _salvar_agendamento_devolucao():
119         agendamento_emprestimo_id = request.form['agendamento_emprestimo_id']
120         data_agendamento = request.form['data_agendamento']
121         horario = request.form['horario']
122
123         agendamento_original = Agendamento.query.get_or_404(agendamento_emprestimo_id)
124
125         if agendamento_original.aluno_id != session.get('usuario_id'):
```

Fonte: Desenvolvido pelos autores (2025) – GitHub

Este arquivo contém a lógica mais importante do sistema: a transação de empréstimos através de agendamentos feitos pelos alunos e confirmados pelo bibliotecário referente à unidade do livro solicitado.

O código demonstra o fluxo de trabalho de um empréstimo, alterando o status do agendamento e definindo a data de devolução prevista automaticamente. Disponibilizando depois funções como devolução e renovações. Ou seja, o arquivo integra as funções dos dois tipos de usuários.

Figura 30 – Sistema de Relatórios

```

76     # Formatando os dados para JSON
77     data = {
78         'emprestimos_15_dias': [{data: str(e.data), 'total': e.total} for e in emprestimos_15_dias],
79         'devolucoes_15_dias': [{data: str(d.data), 'total': d.total} for d in devolucoes_15_dias],
80         'top_5_livros': [{'titulo': t.titulo, 'total': t.total_emprestimos} for t in top_5_livros],
81         'generos_mais_emprestados': [{'genero': g.genero, 'total': g.total_emprestimos} for g in generos_mais_emprestados]
82     }
83
84     return data
85
86     @relatorios_bp.route('/bibliotecario/relatorios')
87     def relatorios():
88         if not session.get('logged_in') or session.get('tipo_usuario') != 'Bibliotecario':
89             flash('Acesso negado! Faça login como bibliotecário.', 'error')
90             return redirect(url_for('login'))
91
92         # Filtrar relatórios por unidade
93         unidade_bibliotecario = session.get('unidade')
94

```

Fonte: Desenvolvido pelos autores (2025) – GitHub

Este arquivo é crucial para demonstrar que o sistema não é apenas transacional, mas também gera valor através de dados. Este trecho do código mostra a capacidade de realizar consultas complexas para fins de análise do funcionamento da biblioteca.

Demonstra consultas avançadas através do uso de funções do SQLAlchemy (func.date, func.count, group_by, join) para extrair informações gerenciais que vão além de um simples CRUD. Também é a base para a criação de gráficos e tabelas utilizando a biblioteca de JavaScript, Chart.js e comunicando com um arquivo JSON (JavaScript Object Notation) responsável pela troca de dados e transferência de informações entre um servidor e um aplicativo web.

Figura 31 – Estrutura de Layout e Responsividade

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>{% block title %}Sistema Biblioteca{% endblock %}</title>
7
8      <!-- Bootstrap -->
9      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/lux/bootstrap.min.css" rel="stylesheet">
10
11     <!-- Boxicons -->
12     <link href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" rel="stylesheet">
13
14     <!-- Bootstrap Icons -->
15     <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css" rel="stylesheet">
16
17     <!-- CSS Personalizado -->
18     <link rel="stylesheet" href="{{ url_for('static', filename='css/base.css') }}">
19
```

Fonte: Desenvolvido pelos autores (2025) – GitHub

Para demonstrar uma parte da aplicação mobile, o foco da programação foi no layout adaptado a diferentes dispositivos. Este trecho mostra a fundação do front-end, o uso de templates e a inclusão dos componentes de navegação.

3 METODOLOGIA

Nesta seção, são apresentadas as metodologias e técnicas utilizadas no desenvolvimento do projeto. Aborda-se o uso das metodologias ágeis, com destaque para o Scrum, que orientou o gerenciamento das atividades. Em seguida, descreve-se o levantamento de requisitos, englobando os funcionais e não funcionais, bem como as técnicas aplicadas, como entrevistas e brainstorming. Por fim, trata-se da priorização dos requisitos e da aplicação do Gráfico de Gantt para o planejamento e acompanhamento das etapas do projeto.

3.1 Metodologia ágil

A metodologia ágil é uma forma de gerir projetos, em que se busca a otimização dos processos. Em vez de seguir um plano rígido — como na gestão tradicional —, na metodologia ágil, podemos fazer ajustes e melhorias ao longo do desenvolvimento de projetos (ALURA, 2019). Isso significa identificar e corrigir falhas rapidamente, garantindo resultados mais efetivos. Portanto, essa forma de gestão permite às pessoas gerentes uma abordagem moderna e eficiente para enfrentar desafios do mercado.

A utilização da agilidade promove uma cultura de colaboração e engajamento da equipe, pois a comunicação entre as pessoas do projeto é facilitada, o que permite a troca de ideias e o alinhamento constante dos objetivos. O foco está no trabalho em equipe e na entrega de valor ao cliente.

3.1.1 Metodologia ágil Scrum

A metodologia ágil aplicada em nosso projeto é a chamada metodologia Scrum. O Scrum é um framework ágil que visa facilitar a colaboração, a adaptação e a entrega de forma mais flexível para lidar com mudanças e busca maximizar o valor entregue ao cliente (AWS, s.d.). Essa metodologia ágil divide o trabalho em iterações chamadas de Sprints, que são acompanhadas por cerimônias como as reuniões de planejamento, revisão e retrospectiva.

As funções no Scrum são desempenhadas por papéis específicos dentro da equipe, cada um com responsabilidades claras que garantem o bom andamento do processo. Esses papéis são:

- **Scrum Master:** atua como facilitador, garantindo que o framework seja seguido e ajudando a remover impedimentos que possam atrapalhar a equipe.
 - Planejar os recursos necessários para cada Sprint.
 - Promover outros eventos de Sprint e reuniões de equipe.
 - Conduzir a transformação digital na equipe.
 - Promover treinamentos da equipe ao adotar novas tecnologias.
 - Comunicar-se com grupos externos para solucionar possíveis desafios que a equipe esteja enfrentando como um todo.
- **Product Owner (PO):** responsável por maximizar o valor do produto, definindo e priorizando os itens do Product Backlog.
 - Fornecem à equipe uma orientação clara sobre quais recursos entregarão em seguida.
 - Fazem a ponte entre o que a empresa quer e o que a equipe entende.
 - Decidem quando e com que frequência os resultados serão apresentados.
- **Time de Desenvolvimento:** grupo multidisciplinar responsável por transformar as necessidades priorizadas em incrementos de produto prontos ao final de cada Sprint.
 - Trabalham de maneira colaborativa para garantir a conclusão eficaz do Sprint.
 - Defendem práticas de desenvolvimento sustentável.
 - São auto-organizadas e efetuam os projetos com um senso de coletividade evidente.
 - Orientam o planejamento e a estimativa de quanto trabalho conseguem concluir em cada Sprint.

O ciclo do Scrum tem início com o Planejamento de Sprint, reunião em que a equipe define os objetivos do próximo ciclo, estima o trabalho a ser realizado e estabelece metas específicas e alcançáveis. Em seguida, acontece o Sprint, período

em que a equipe trabalha de forma colaborativa para desenvolver um incremento funcional do produto. Ao final do Sprint, realiza-se a Revisão de Sprint, uma sessão em que a equipe apresenta o trabalho concluído às partes interessadas.

3.2 Levantamento de requisitos

O levantamento de requisitos representa uma das etapas mais críticas do processo de desenvolvimento de software, pois fornece a base para todas as fases subsequentes (CEDRO, 2018). Trata-se do momento em que são identificadas, documentadas e validadas as necessidades dos usuários e das partes interessadas, garantindo que o sistema em desenvolvimento esteja alinhado com os objetivos do negócio. Esse processo visa reduzir incertezas, evitar ambiguidades e assegurar que o software atenda às demandas reais do cliente.

Os requisitos de um sistema podem ser classificados em duas categorias principais. Os requisitos *funcionais* descrevem as funcionalidades que o sistema deve executar, ou seja, as ações e comportamentos que a aplicação precisa oferecer para atender às necessidades do usuário. Já os requisitos *não funcionais* especificam restrições e atributos de qualidade relacionados ao sistema.

3.2.1 Requisitos Funcionais

Os requisitos funcionais (RFs) são diretrizes específicas que descrevem os comportamentos, funções e operações que um software ou sistema deve executar (VISURE SOLUTIONS, 2025). No desenvolvimento de software e sistemas, esses requisitos descrevem o que o sistema deve fazer para atender às necessidades do usuário e aos objetivos de negócios. Os requisitos funcionais são normalmente detalhados na documentação, como uma especificação de requisitos, para garantir clareza e alinhamento entre equipes de desenvolvimento, partes interessadas e usuários finais.

No desenvolvimento de software e sistemas, os requisitos funcionais definem as funções ou ações essenciais que um sistema deve executar. Eles podem variar de interações do usuário, processamento de dados, cálculos e integrações com outros sistemas, até as etapas envolvidas na conclusão de um fluxo de trabalho. Eles diferem dos requisitos não funcionais, que se concentram em atributos do sistema como

desempenho, usabilidade e segurança. Os RFs respondem ao "o quê" e "como" do comportamento de um sistema para fornecer um modelo para desenvolvimento e teste.

Tabela 1 – Requisitos funcionais

IDENTIFICADOR	NOME	DESCRÍÇÃO	DEPENDÊNCIA
RF01	Cadastrar Usuário	O sistema deve permitir o cadastro de usuários, diferenciando alunos e bibliotecários, registrando nome completo, e-mail, senha e unidade escolar. O sistema deve garantir que cada e-mail seja único e que todos os campos obrigatórios sejam preenchidos.	—
RF02	Realizar Login	Alunos e bibliotecários devem poder acessar o sistema utilizando e-mail e senha válidos. O sistema deve validar as credenciais e impedir acesso em caso de dados incorretos.	RF01
RF03	Visualizar Página Inicial	Após o login, o usuário deve acessar sua página inicial, que apresenta funcionalidades específicas de acordo com seu perfil: Aluno: acesso ao catálogo de livros, agendamento de empréstimo/devolução e histórico de empréstimos. Bibliotecário: acesso à lista de agendamentos, registro de empréstimos e devoluções, e relatórios de movimentação de livros.	RF02
RF04	Visualizar Catálogo de Livros	O aluno deve poder pesquisar livros pelo título, autor, categoria ou palavra-chave, visualizando disponibilidade em cada unidade escolar.	RF03

RF05	Agendar Empréstimo e Devolução de Livro	O aluno pode selecionar um livro disponível e a unidade escolar para retirada, agendando o horário de empréstimo ou devolução. O bibliotecário da unidade recebe notificação do agendamento e, ao entregar o livro, registra como “Emprestado” ou atualiza o status para “Devolvido”.	RF04
RF06	Renovação de Empréstimo	O aluno pode solicitar até três renovações de um empréstimo, desde que o livro ainda esteja dentro do prazo de devolução. O sistema verifica automaticamente a elegibilidade e confirma a renovação.	RF05
RF07	Histórico de Empréstimos	O aluno deve poder consultar seu histórico completo na “Minha Estante” de empréstimos e devoluções, incluindo datas, status atual e quantidade de renovações utilizadas.	RF05
RF08	Notificações e Alertas	O sistema deve enviar notificações aos alunos sobre prazos de devolução, confirmações de agendamento e disponibilidade de livros reservados, e alertas aos bibliotecários sobre agendamentos e devoluções pendentes.	RF05 / RF06
RF09	Relatórios de Empréstimos e Devoluções	O bibliotecário deve gerar relatórios de movimentação de livros, incluindo número de empréstimos, devoluções, livros mais populares e histórico por unidade escolar ou categoria.	RF05 / RF06
RF10	Gerenciamento de Usuários e Livros	O bibliotecário deve poder: Gerenciar Usuários (alunos): visualizar lista de alunos cadastrados, editar dados (nome, e-mail, unidade), ativar/desativar contas e redefinir senhas.	RF01 / RF03

		Gerenciar Livros: cadastrar novos livros, atualizar informações (título, autor, categoria, unidade, quantidade disponível), alterar status (disponível, emprestado, reservado) e remover exemplares quando necessário.	
RF11	Avaliar Livros	O aluno deve poder avaliar os livros após a devolução, atribuindo uma nota de 1 a 5 estrelas e, opcionalmente, deixar um comentário. As avaliações devem ser exibidas na página de detalhes de cada livro.	RF05 / RF07
RF12	Recomendações de Livros por Gênero	O sistema deve exibir recomendações personalizadas de livros com base no histórico de empréstimos e nos gêneros mais acessados pelo aluno, sugerindo obras semelhantes.	RF07 / RF11
RF13	Escanear Código de Barras do Livro	O sistema deve permitir que o usuário utilize a câmera do dispositivo para escanear o código de barras de um livro e, automaticamente, exibir as informações correspondentes no catálogo.	RF04
RF14	Carteirinha Digital da Biblioteca	O sistema deve gerar uma carteirinha digital para cada aluno cadastrado, contendo nome, número de matrícula e QR Code de identificação, que pode ser utilizada para validação rápida em empréstimos e devoluções.	RF01 / RF05
RF15	Barra de Pesquisa Aprimorada e Categorias por Gênero	O sistema deve permitir a busca rápida de livros por meio de uma barra de pesquisa inteligente e organizado em categorias por gênero facilitando a navegação.	RF04

Fonte: Elaborado pelos autores (2025)

3.2.2 Requisitos Não Funcionais

Os requisitos não funcionais (RNFs) especificam os padrões e qualidades que um sistema deve atender para ter um desempenho eficaz, com foco em *como* um sistema opera em vez de *o que sim* (VISURE SOLUTIONS, 2025). Enquanto requisitos funcionais definem tarefas ou comportamentos específicos de um sistema de software, os requisitos não funcionais descrevem qualidades como velocidade, confiabilidade e segurança que ditam o desempenho, usabilidade e resiliência do sistema.

No desenvolvimento de software, os RNFs desempenham um papel crucial ao fornecer uma referência para a qualidade do sistema e experiência do usuário. Sem eles, mesmo um software rico em recursos pode deixar de atender às expectativas se não tiver velocidade, escalabilidade ou facilidade de uso. Ao abordar os requisitos não funcionais cedo e completamente, as equipes de desenvolvimento aumentam não apenas a estabilidade e a eficiência do sistema, mas também a satisfação e a confiança de seus usuários.

Tabela 2 – Requisitos não funcionais

IDENTIFICADOR	NOME	DESCRIÇÃO
RNF01	Usabilidade	O sistema deve possuir uma interface simples, intuitiva e responsiva, permitindo que usuários de diferentes níveis de familiaridade tecnológica consigam utilizá-lo sem dificuldades.
RNF02	Desempenho	O sistema deve processar operações de login, cadastro, busca e empréstimo em no máximo 3 segundos, garantindo agilidade no uso diário.
RNF03	Responsividade	O sistema deve ser totalmente responsivo, adaptando sua interface e funcionalidades para uso em dispositivos móveis, tablets e computadores, mantendo boa usabilidade em todos os tamanhos de tela.
RNF04	Segurança de Dados	Todas as senhas de usuários devem ser armazenadas de forma criptografada, e as comunicações entre cliente e servidor devem utilizar protocolo seguro (HTTPS).

RNF05	Disponibilidade	O sistema deve estar disponível para acesso 24 horas por dia, 7 dias por semana, com tempo de inatividade planejado apenas para manutenções agendadas.
RNF06	Escalabilidade	O sistema deve permitir expansão para atender múltiplas unidades escolares sem comprometer o desempenho.
RNF07	Confiabilidade	O sistema deve garantir a integridade das informações de usuários e livros, mesmo em casos de falhas ou interrupções de conexão.
RNF08	Manutenibilidade	O código-fonte do sistema deve ser estruturado de forma modular, facilitando futuras atualizações e correções.
RNF09	Privacidade	O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados (LGPD), assegurando que informações pessoais dos usuários não sejam compartilhadas sem consentimento.
RNF10	Backup Automático	O sistema deve realizar backups automáticos diários do banco de dados, garantindo a recuperação das informações em caso de falhas.

Fonte: Elaborado pelos autores (2025)

3.3 Técnicas de Levantamento de Requisitos

O levantamento de requisitos é uma etapa fundamental no desenvolvimento de software, pois define as necessidades e expectativas dos usuários e orienta a construção do sistema. Para realizar esse processo de maneira eficiente, diversas técnicas podem ser utilizadas, cada uma com suas características, vantagens e limitações.

Para o desenvolvimento deste projeto, foram escolhidas as técnicas de *entrevista* e *brainstorming*, combinando a coleta detalhada de informações com a geração colaborativa de ideias. Essa abordagem, envolvendo os *stakeholders* relevantes, permite um levantamento de requisitos mais completo, assegurando que as funcionalidades e atributos de qualidade do sistema estejam alinhados às necessidades reais dos usuários.

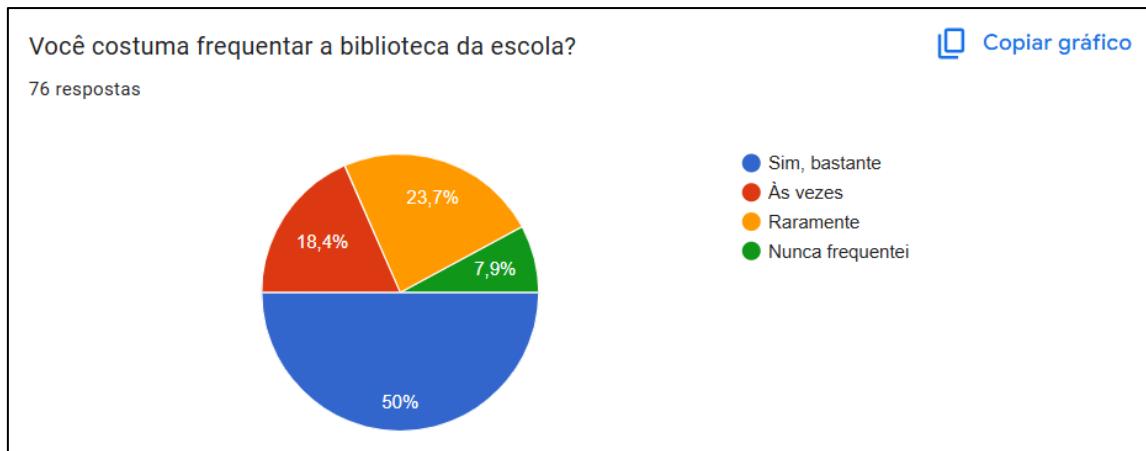
3.3.1 Entrevista

A entrevista é uma técnica que consiste em reuniões com os stakeholders, formais ou informais, nas quais são colocadas questões formuladas pela equipe de engenharia de requisitos sobre o processo de trabalho atual do cliente e sobre o sistema que será desenvolvido (ESTRATÉGIA, 2023). A entrevista está presente na maioria dos processos de elaboração de requisitos, pois é a partir dela que podem surgir necessidades para a utilização de outras técnicas.

No contexto deste projeto, a técnica de entrevista foi aplicada com *alunos* e *bibliotecários*, permitindo à equipe compreender detalhadamente como cada grupo utiliza o espaço de ensino e a biblioteca, bem como identificar necessidades e dificuldades enfrentadas no dia a dia. Para facilitar a coleta de respostas, o formulário da entrevista foi elaborado utilizando a ferramenta Google Forms e compartilhado por meio de canais de comunicação entre os alunos de cada rede do SESI-SENAI, possibilitando maior alcance e participação dos alunos de diferentes unidades.

FORMULÁRIO DIRECIONADO AOS ALUNOS

Gráfico 1 – Você costuma frequentar a biblioteca da escola?



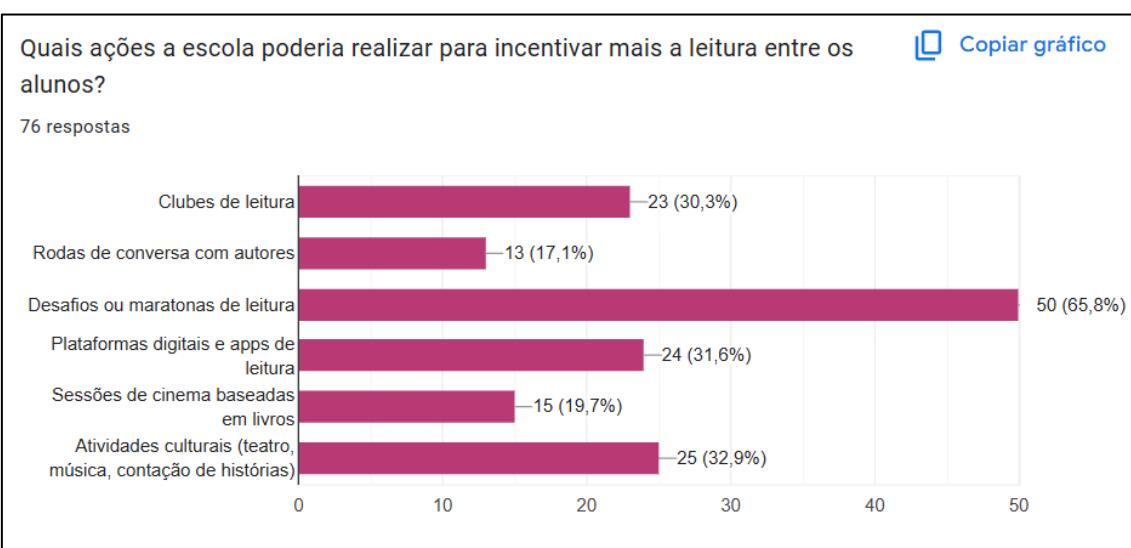
Fonte: Autores (2025) | Google Forms

Gráfico 2 – Você se sente incentivado(a) a leitura pela escola?



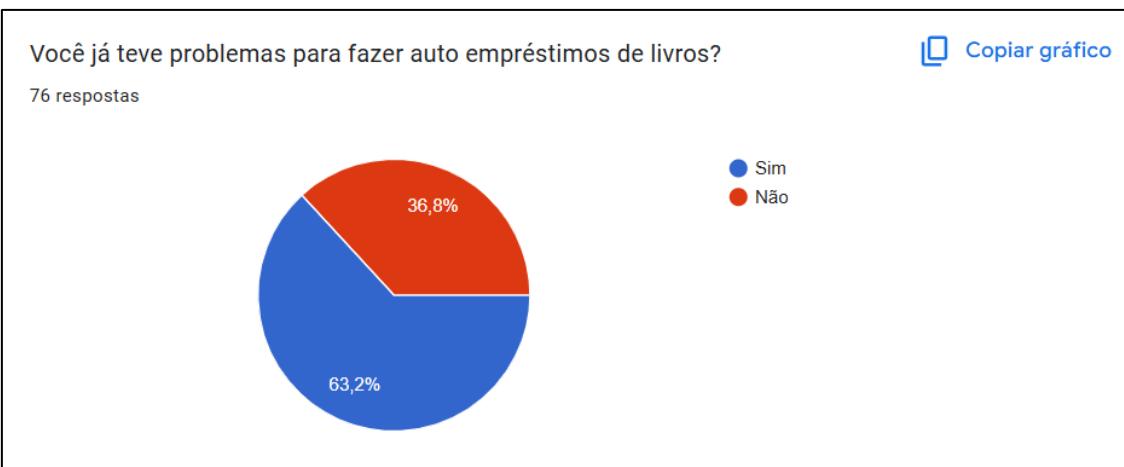
Fonte: Autores (2025) | Google Forms

Gráfico 3 – Quais ações a escola poderia realizar para incentivar mais a leitura entre os alunos?



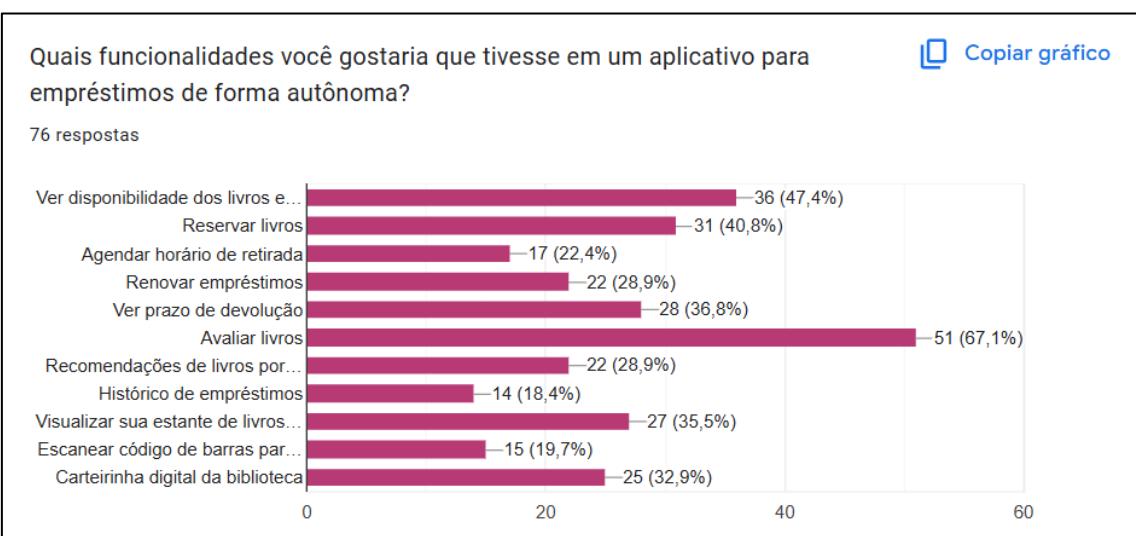
Fonte: Autores (2025) | Google Forms

Gráfico 4 – Você já teve problemas para fazer auto empréstimos de livros?



Fonte: Autores (2025) | Google Forms

Gráfico 5 – Quais funcionalidades você gostaria que tivesse em um aplicativo para empréstimos de forma autônoma?



Fonte: Autores (2025) | Google Forms

3.3.2 Brainstorming

O brainstorming é uma técnica que consiste em uma ou várias reuniões em que os participantes sugerem e exploram ideias de forma livre, sem críticas ou julgamentos (MIRO, s.d.). Seu objetivo principal é a geração de ideias, seguindo as etapas:

- **Etapa 1 – Objetivo Principal:** Definir a meta do brainstorming. Identificar o problema ou oportunidade a ser explorado.

- **Etapa 2 – Ideias Iniciais:** Anotar todas as ideias que vem à mente, sem julgamentos. Listar sugestões espontâneas de todos os participantes.
- **Etapa 3 – Pesquisa e Referências:** Coletar dados relevantes e antecedentes. Analisar estudos de caso ou exemplos relacionados.
- **Etapa 4 – Categorias e Temas:** Agrupar ideias semelhantes em categorias. Identificar temas recorrentes e subtemas.
- **Etapa 5 – Avaliação e Seleção:** Avaliar a viabilidade e impacto das ideias. Priorizar ideias que melhor atendem aos critérios definidos.
- **Etapa 6 – Plano de Ação:** Desenvolver um cronograma e etapas para implementação. Designar responsabilidades e recursos necessários.

No contexto deste projeto, o brainstorming foi aplicado em reuniões com os membros da equipe, onde discutimos quais áreas poderiam ser melhoradas com o desenvolvimento do nosso trabalho. Após a análise, identificamos que seria mais prático, considerando nosso papel como alunos, focar na melhoria do local de ensino, por isso decidimos otimizar o sistema bibliotecário das escolas. Essa escolha permitiu que levantássemos pontos relevantes e identificássemos oportunidades de aprimoramento que impactariam diretamente a experiência de aprendizado.

O brainstorming teve como objetivo identificar oportunidades de melhoria no ambiente escolar por meio de soluções tecnológicas. Foram discutidos problemas e necessidades do cotidiano dos alunos e funcionários, resultando em ideias como um sistema bibliotecário, uma secretaria virtual e aplicativos de apoio ao ensino.

3.4 Priorização de requisitos

A priorização de requisitos é uma etapa fundamental no processo de engenharia de software, pois auxilia na definição da ordem em que as funcionalidades devem ser implementadas, considerando o valor de negócio, custo, riscos e restrições técnicas e garantindo entregas com impacto real e maior aceitação do produto.

Os stakeholders exercem papel central nesse processo, pois são os responsáveis por expressar suas necessidades, expectativas e restrições. No caso de um sistema de gestão de bibliotecas, por exemplo, os principais stakeholders são **os alunos** — que utilizam o sistema para consultar o catálogo, reservar livros e

acompanhar empréstimos — e os **bibliotecários**, que gerenciam usuários, controlam empréstimos e devoluções e administram o acervo. Dessa forma, a priorização não apenas orienta o desenvolvimento técnico, mas também assegura o alinhamento entre as metas e a experiência dos usuários finais.

3.4.1 *Tipo de priorização de requisitos*

Existem diversas técnicas para priorização. Uma das mais utilizadas é o método **MoSCoW**, que classifica os requisitos em quatro categorias: *Must have* (essenciais), *Should have* (importantes), *Could have* (desejáveis) e *Won't have* (não serão implementados no momento).

Nosso projeto utilizou o método MoSCoW para definir a prioridade de cada requisito no sistema. Abaixo, apresentamos a tabela com os requisitos classificados conforme essa técnica.

Tabela 3 – Requisitos: *Must have* (essenciais)

MUST HAVE		
Identificador	Nome	Descrição
RF01	Cadastrar Usuário	Essencial para permitir o acesso de usuários ao sistema e diferenciar perfis (alunos e bibliotecários).
RF02	Realizar Login	Fundamental para autenticação e controle de acesso ao sistema.
RF03	Visualizar Página Inicial	Base da navegação; direciona o usuário às funcionalidades principais.
RF04	Visualizar Catálogo de Livros	Principal funcionalidade para busca e seleção de livros no sistema.
RF05	Agendar Empréstimo e Devolução de Livro	Recurso central para operação da biblioteca digital.
RF06	Renovação de Empréstimo	Funcionalidade conveniente para alunos, também é essencial para operação do sistema.
RF10	Gerenciamento de Usuários e Livros	Essencial para controle administrativo e manutenção do acervo.

RNF01	Usabilidade	Essencial para que todos os usuários consigam utilizar o sistema sem dificuldades.
RNF02	Desempenho	Necessário para garantir agilidade no uso diário e evitar frustração do usuário.
RNF03	Responsividade	Fundamental para permitir acesso em diferentes tipos de dispositivos.
RNF04	Segurança de Dados	Essencial para proteger dados dos usuários e garantir confiança no sistema.
RNF07	Confiabilidade	Essencial para a confiança no sistema e consistência das informações.
RNF09	Privacidade	Obrigatório legalmente e essencial para a segurança dos usuários.

Fonte: Elaborado pelos autores (2025)

Tabela 4 – Requisitos: *Should have* (importantes)

SHOULD HAVE		
Identificador	Nome	Descrição
RF09	Relatórios de Empréstimos e Devoluções	Importante para gestão da biblioteca, mas o sistema pode funcionar sem inicialmente.
RF14	Carteirinha Digital da Biblioteca	Útil para modernizar operações, mas não é crítico para o funcionamento.
RF15	Barra de Pesquisa Aprimorada e Categorias por Gênero	Facilita a navegação e a busca pelo catálogo, mas não é essencial para operação do sistema.
RNF05	Disponibilidade	Importante para garantir continuidade do serviço, mas pequenas interrupções planejadas são aceitáveis.
RNF06	Escalabilidade	Necessário para expansão futura, mas não crítico na versão inicial.
RNF08	Manutenibilidade	Facilita manutenção futura, mas não impede o funcionamento inicial do sistema.

RNF10	Backup Automático	Garante recuperação de dados em caso de falhas, importante, mas não impede uso diário se falhar eventualmente.
--------------	--------------------------	--

Fonte: Elaborado pelos autores (2025)

Tabela 5 – Requisitos: *Could have* (desejáveis)

COULD HAVE		
Identificador	Nome	Descrição
RF07	Histórico de Empréstimos	Permite acompanhamento dos empréstimos, mas não crítico para funcionamento principal.
RF08	Notificações e Alertas	Facilita acompanhamento de prazos e confirmações, mas não impede uso do sistema.

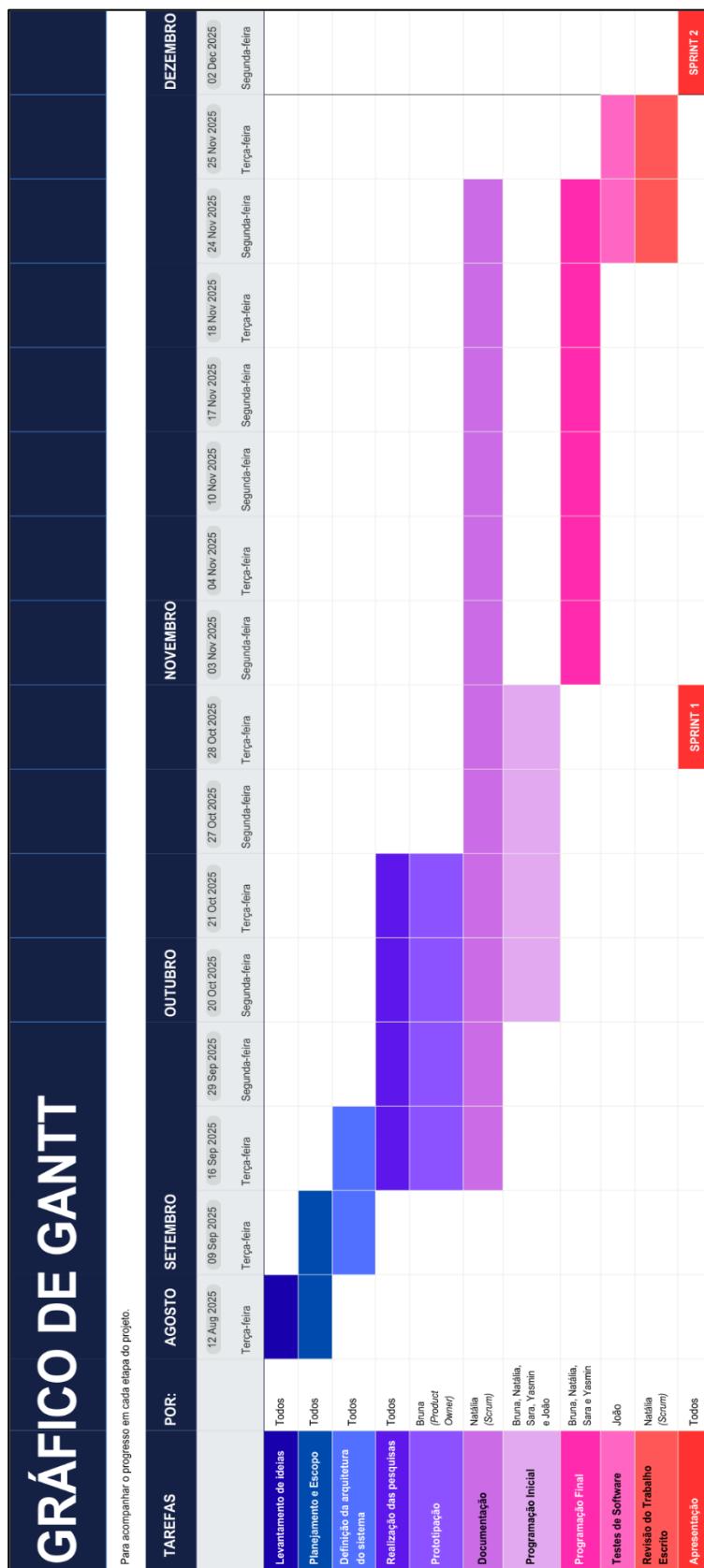
Fonte: Elaborado pelos autores (2025)

Tabela 6 – Requisitos: *Won't have* (não serão implementados no momento)

WON'T HAVE		
Identificador	Nome	Descrição
RF11	Avaliar Livros	Funcionalidade interessante, mas complexa para implementar nesta fase do projeto.
RF12	Recomendações de Livros por Gênero	Exige algoritmos avançados, não será implementado agora.
RF13	Escanear Código de Barras do Livro	Recurso opcional, não será desenvolvido nesta versão do projeto.

Fonte: Elaborado pelos autores (2025)

3.5 Gráfico de Gantt



Fonte: Elaborado pelos autores (2025)

3.6 Modelagem Prática do Sistema

Este tópico apresenta a modelagem prática do sistema de gerenciamento de bibliotecas, detalhando a estrutura e o funcionamento por meio de dois diagramas fundamentais: o Diagrama Entidade-Relacionamento (DER) e o Diagrama de Casos de Uso. O DER descreve as entidades principais — usuários, livros, agendamentos e logins — e seus relacionamentos, garantindo integridade e eficiência na gestão de dados. Já o Diagrama de Casos de Uso ilustra as interações dos atores (aluno e bibliotecário) com o sistema, destacando funcionalidades como empréstimos, renovações, cadastro de livros e geração de relatórios. Essas modelagens são essenciais para orientar o desenvolvimento e assegurar que o sistema atenda aos requisitos de forma clara e organizada.

3.6.1 Aplicação do diagrama de caso de uso

Para modelar a interação entre os usuários e o sistema, foi utilizado o Diagrama de Casos de Uso, uma ferramenta da UML (Unified Modeling Language) que permite representar as funcionalidades do sistema sob a perspectiva de quem o utiliza.

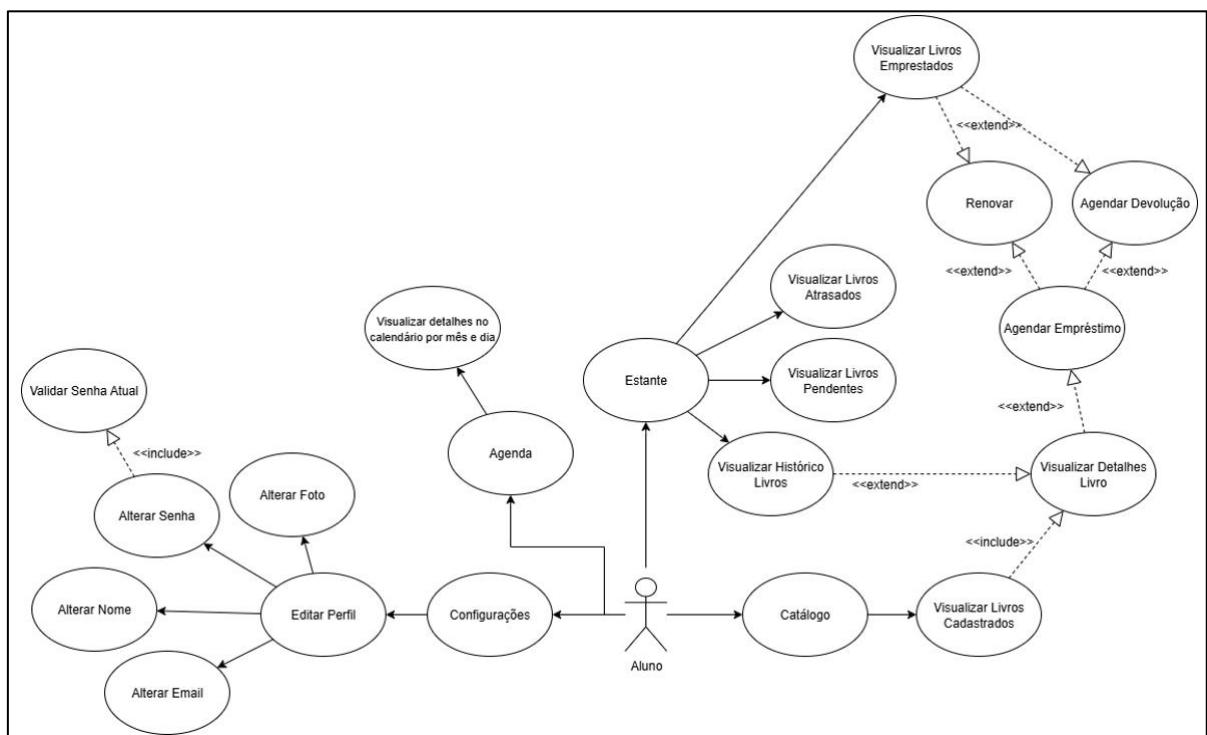
Conforme a teoria de modelagem de sistemas, o diagrama de caso de uso resume os detalhes dos usuários (atores) e as interações que eles podem realizar. Ele oferece uma visão geral do relacionamento entre casos de uso, atores e o sistema, descrevendo os cenários e as metas que o aplicativo ajuda seus usuários a atingirem, além de delimitar o escopo do sistema.

No contexto do sistema de biblioteca, foram identificados dois atores principais: o **Aluno** e o **Bibliotecário**. Cada um possui um conjunto distinto de permissões e interações, que são detalhadas a seguir.

ALUNO

O Aluno representa o usuário final do sistema, cujo principal objetivo é a busca e o empréstimo de livros. O diagrama a seguir (Figura 2) ilustra as funcionalidades disponíveis para este ator.

Figura 24 – Diagrama de caso de uso do Aluno



Fonte: Elaborado pelos autores (2025) | Draw.io

As principais funcionalidades (casos de uso) para o Aluno são:

Gerenciar Perfil e Configurações:

- Editar Perfil:** O aluno pode alterar seus dados cadastrais, como Alterar Nome, Alterar Email e Alterar Foto.
- Alterar Senha:** Para garantir a segurança da conta, o aluno pode modificar sua senha de acesso. Este caso de uso inclui um processo de validação em duas etapas: primeiro, o sistema verifica se a senha atual fornecida pelo usuário está correta; segundo, assegura que a nova senha seja diferente da antiga. Este procedimento garante que a alteração seja realizada pelo titular da conta e promove a utilização de novas credenciais, fortalecendo a segurança.

Interagir com o Acervo:

- **Catálogo:** Permite ao aluno Visualizar Livros Cadastrados no acervo das bibliotecas disponíveis. A partir desta visualização, ele pode Visualizar Detalhes do Livro para obter mais informações sobre uma obra específica.
- **Estante:** Funcionalidade que centraliza a gestão dos livros relacionados ao aluno, permitindo Visualizar Livros Emprestados, Livros Atrasados, Livros Pendentes e o Histórico de Livros já lidos.

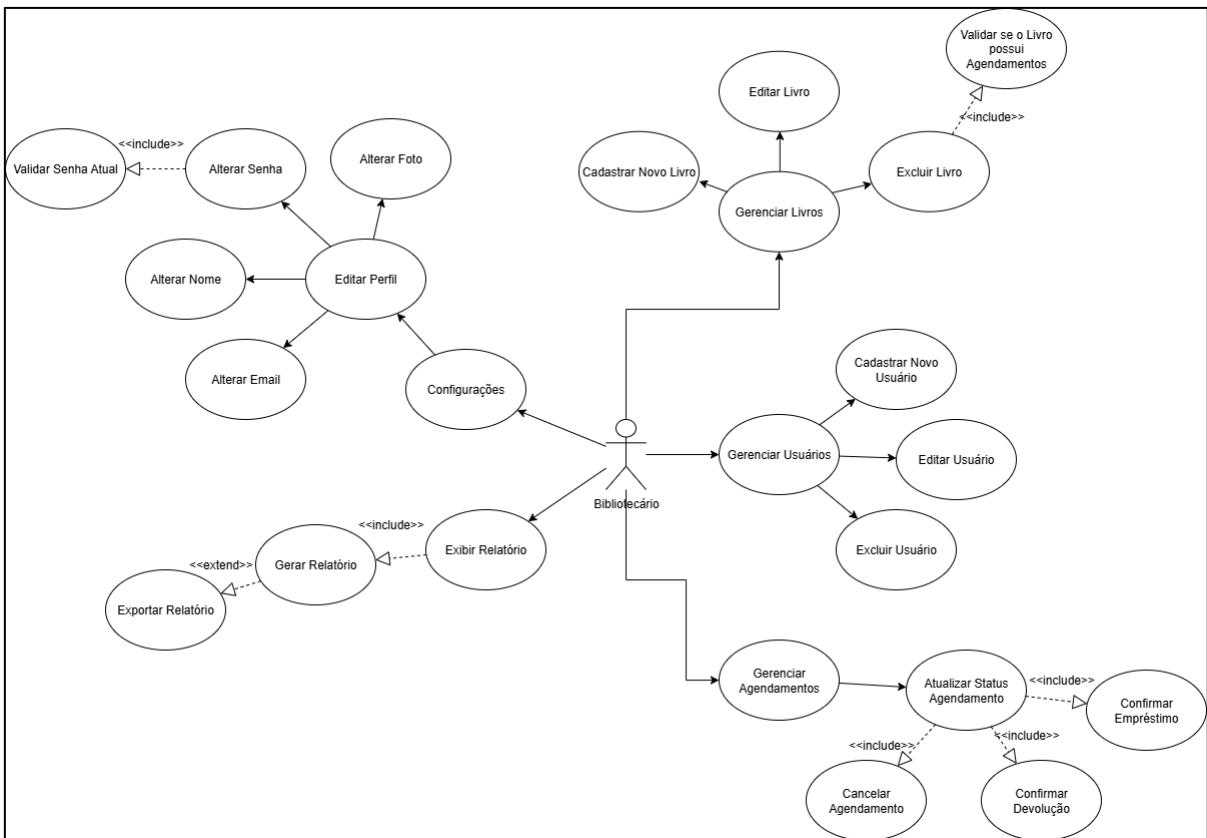
Gerenciar Empréstimos e Agendamentos:

- **Agendar Empréstimo:** A partir da visualização de detalhes de um livro, o aluno pode solicitar o empréstimo da obra.
- **Renovar:** Caso de uso que estende a funcionalidade de Visualizar Livros Emprestados, permitindo ao aluno solicitar a prorrogação do prazo de devolução.
- **Agendar Devolução:** Funcionalidade que também estende a visualização de livros emprestados, permitindo ao aluno formalizar a intenção de devolver uma obra.
- **Agenda:** Permite ao aluno visualizar detalhes no calendário por mês e dia, facilitando o acompanhamento de prazos e agendamentos.

BIBLIOTECÁRIO

O Bibliotecário é o ator responsável pela administração do sistema, gerenciando o acervo, os usuários e as operações de empréstimo. Suas interações (Figura 3) refletem um perfil com privilégios elevados.

Figura 25 – Diagrama de caso de uso do Bibliotecário



Fonte: Elaborado pelos autores (2025) | Draw.io

As principais funcionalidades (casos de uso) para o Bibliotecário são:

- **Gerenciar Livros:** O bibliotecário tem controle total sobre o acervo, podendo Cadastrar Novo Livro, Editar Livro existente e Excluir Livro. A exclusão de um livro inclui uma verificação crucial: Validar se o Livro possui Agendamentos, impedindo a remoção de obras que estejam em processo de empréstimo.
- **Gerenciar Usuários:** Responsável pela administração das contas de usuário, o bibliotecário pode Cadastrar Novo Usuário, Editar Usuário e Excluir Usuário.
- **Gerenciar Agendamentos:** Atua como o operador das transações da biblioteca.
 - **Atualizar Status Agendamento:** Caso de uso central que inclui as ações de Confirmar Empréstimo e Confirmar Devolução, efetivando as operações no sistema.
 - **Cancelar Agendamento:** Permite invalidar uma solicitação de empréstimo antes de sua efetivação.

- **Gerar Relatórios:** Para fins administrativos e de controle, o bibliotecário pode Exibir Relatório. Este caso de uso inclui a funcionalidade de Gerar Relatório e estende a possibilidade de Exportar Relatório para formatos externos (PDF).
- **Gerenciar Perfil:** Assim como o aluno, o bibliotecário também pode editar suas próprias informações de perfil e senha.

A definição clara desses casos de uso foi fundamental para guiar o desenvolvimento das interfaces e da lógica de negócio, garantindo que todos os requisitos funcionais do sistema fossem atendidos de maneira organizada e rastreável.

ELEMENTOS DO DIAGRAMA DE CASO DE USO

São componentes comuns:

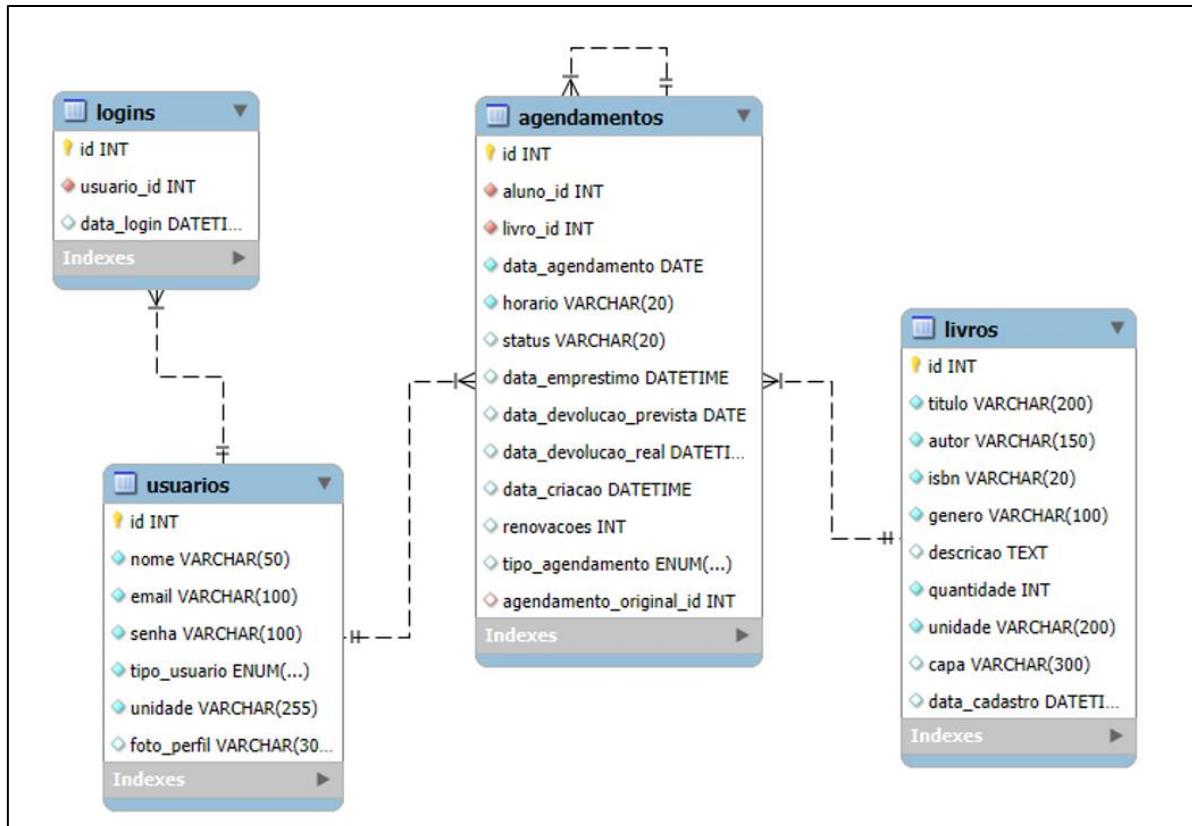
- **Atores:** os usuários que interagem com o sistema. Ator pode ser uma pessoa, organização ou sistema externo que interage com seu aplicativo ou sistema. Eles devem ser objetos externos que produzam ou consumam dados.
- **Sistema:** uma sequência específica de ações e interações entre os atores e o sistema. O sistema também pode ser chamado de cenário.
- **Metas:** o resultado final da maioria dos casos de uso. Um diagrama criado corretamente deve descrever as atividades e variantes usadas para atingir a meta.

3.6.2 Aplicação do diagrama entidade-relacionamento (DER)

O desenvolvimento do sistema de *gerenciamento de bibliotecas* foi fundamentado em uma estrutura de banco de dados relacional, projetada para garantir a integridade, a consistência e a eficiência no armazenamento e na recuperação das informações. A arquitetura de dados é composta por quatro entidades principais: usuários, livros, agendamentos e logins.

O diagrama entidade-relacionamento (DER) a seguir ilustra a estrutura e as interconexões entre essas tabelas.

Figura 26 – Diagrama entidade-relacionamento



Fonte: Elaborado pelos autores (2025) | MySQL Workbench

A seguir, detalha-se a função de cada tabela e seus respectivos atributos.

TABELA USUÁRIOS

A tabela usuários é a entidade central para o armazenamento de dados cadastrais de todos os indivíduos que interagem com o sistema, incluindo alunos e bibliotecários:

- Id (chave primária):** Identificador numérico único para cada usuário, garantindo a unicidade dos registros.
- nome, email, senha:** Atributos para identificação, comunicação e autenticação do usuário. A senha é armazenada de forma criptografada para assegurar a segurança das credenciais.

- **tipo_usuario:** Campo do tipo ENUM que categoriza o usuário ("aluno", "bibliotecário"), sendo fundamental para o controle de permissões e níveis de acesso às funcionalidades do sistema.
- **unidade, foto_perfil:** Campos adicionais para especificar a unidade de vínculo do usuário e armazenar a referência para sua imagem de perfil.

TABELA LIVROS

Esta tabela funciona como o catálogo digital da biblioteca, contendo o acervo de todas as obras disponíveis:

- **id (chave primária):** Identificador numérico exclusivo para cada título de livro.
- **titulo, autor, isbn, genero, descricao:** Dados essenciais que descrevem a obra, permitindo sua correta catalogação e facilitando a busca por parte dos usuários em ferramentas diversas.
- **quantidade:** Atributo numérico que controla o número de exemplares disponíveis para empréstimo.
- **unidade:** Especifica a localização física ou a unidade administrativa à qual o livro pertence.
- **data_cadastro:** Registra a data e hora em que o livro foi inserido no sistema, servindo como um dado de auditoria.

TABELA LOGINS

A tabela logins foi projetada com o propósito de auditoria, registrando o histórico de acessos dos usuários à plataforma:

- **id (chave primária):** Identificador único para cada sessão de login.
- **usuario_id (chave estrangeira):** Estabelece um relacionamento de N para 1 com a tabela usuarios, vinculando cada registro de login a um usuário específico.
- **data_login:** Armazena a data e a hora exatas do acesso, permitindo análises de utilização do sistema.

TABELA AGENDAMENTOS

Esta é a entidade transacional do sistema, responsável por gerenciar todo o ciclo de vida dos empréstimos, devoluções e renovações de livros. Ela atua como uma tabela associativa, conectando as entidades usuários e livros:

- **id (chave primária)**: Identificador único para cada operação de agendamento ou empréstimo.
- **aluno_id (chave estrangeira)**: Referência o id da tabela usuarios, identificando o solicitante do empréstimo.
- **livro_id (chave estrangeira)**: Referência o id da tabela livros, identificando a obra transacionada.
- **data_emprestimo, data_devolucao_prevista, data_devolucao_real**: Conjunto de atributos de data que controlam o período do empréstimo, a data limite para devolução e a data efetiva em que a devolução ocorreu.
- **status**: Campo de estado que descreve a situação atual da transação ("agendado", "emprestado", "devolvido"), essencial para a lógica de negócio.
- **renovacoes**: Contador que limita o número de renovações permitidas para um mesmo empréstimo.
- **agendamento_original_id**: Atributo de auto-relacionamento que vincula uma renovação ao seu registro de empréstimo original, mantendo a rastreabilidade do histórico.

Em síntese, o modelo de dados adotado é robusto, permitindo que o sistema execute suas funções de forma segura e escalável, além de prover uma base sólida para a futura implementação de novas funcionalidades e relatórios gerenciais.

3.7 Testes de Software Aplicados ao Sistema

Para garantir o correto funcionamento do sistema e validar todas as suas funcionalidades, foram aplicados diferentes tipos de testes de software ao longo do

desenvolvimento. Esses testes tiveram como objetivo identificar falhas e confirmar o comportamento esperado dos módulos.

Inicialmente, foram realizados **testes unitários**, que verificaram individualmente funções e componentes isolados do sistema. Nessa fase, foram analisados elementos como funcionamento de botões, chamadas de funções específicas, validação de campos de entrada e respostas do sistema para cada ação.

Em seguida, aplicamos **testes de integração**, que analisaram o comportamento conjunto entre diferentes partes do sistema. Exemplos avaliados incluem:

- Envio de dados do front-end para o banco de dados;
- Salvamento, edição e exclusão de registros;
- Comunicação entre os módulos de agendamento de empréstimos e o painel do bibliotecário;
- Verificação se a solicitação realizada pelo aluno era corretamente transferida e exibida para o bibliotecário aprovar ou rejeitar.

Também foram executados **testes funcionais**, voltados para garantir que cada funcionalidade atendesse exatamente ao que foi planejado nos requisitos do sistema. Além disso, aplicamos testes de usabilidade, observando a facilidade de navegação do usuário, clareza dos botões e organização da interface, garantindo a responsividade e acessibilidade.

Por fim, realizamos testes de sistema, nos quais a aplicação completa foi analisada como um todo, observando estabilidade, fluxo geral de funcionamento e comportamento diante de dados reais. Essa etapa incluiu cenários completos, como realizar um empréstimo, atualizar o acervo, confirmar agendamentos e verificar se todas as notificações e registros eram enviados corretamente para cada usuário envolvido.

CONSIDERAÇÕES FINAIS

Neste trabalho, foi desenvolvido um sistema para bibliotecas, com o objetivo de melhorar os processos administrativos e educativos que facilite o controle de uma biblioteca. Destacamos a relevância do desenvolvimento do Sistema Integrado de Bibliotecas (SiBi) como uma solução tecnológica abrangente, criada para modernizar e otimizar os processos de gestão de uma biblioteca acadêmica.

Ademais, utilizamos um conjunto sólido de linguagens — HTML, CSS, JavaScript e Python — aliado a frameworks e ferramentas como Flask para o backend, MySQL Workbench para definição e manipulação do banco de dados, Flet para a criação de interfaces adicionais e Figma para prototipação, tornou-se possível desenvolver um sistema robusto, coerente e funcional. Para o design da interface, foram incorporados recursos como Bootstrap e Boxicons, garantindo uma experiência visual organizada, responsiva e intuitiva.

Assim, buscamos ir além da simples implementação de funcionalidades, refletindo também sobre o impacto positivo que esse sistema pode gerar no cotidiano acadêmico dos estudantes. O SiBi foi planejado para facilitar o acesso à informação, reduzir filas, diminuir o tempo de atendimento e aumentar a autonomia dos alunos na realização de pesquisas, consultas e agendamentos de empréstimos. Ao oferecer uma navegação clara e organizada, o sistema contribui para um ambiente educacional mais moderno e eficiente, apoiando diretamente o processo de aprendizagem.

Em suma, este trabalho proporcionou uma contribuição significativa para o campo de desenvolvimento de sistemas, fornecendo uma solução com potencial significativo e eficiente para melhorar a relação dos usuários com a biblioteca. Esperamos que as conclusões e recomendações apresentadas inspirem novas pesquisas e aprimoramentos nessa área, contribuindo para o crescimento e sucesso das bibliotecas que buscam se otimizar.

REFERÊNCIAS

- MILANESI, Luís.** *Biblioteca*. Ateliê Editorial, 2002. Disponível em: <https://books.google.com.br/books?id=jxD9My-g1xQC&pg=PA5&hl=pt-BR#v=onepage&q&f=false>.
- ROSA E SILVA, [s.d.] apud PINTO, Júlia; PORTILHO, Gabriela.** *O que são bibliotecas virtuais*. Nova Escola, 2012. Disponível em: <https://novaescola.org.br/conteudo/2226/o-que-sao-bibliotecas-virtuais>.
- CUNHA, Murilo Bastos da.** *Para saber mais: fontes de informação em ciência e tecnologia*. Brasília: Briquet de Lemos/Livros, 2001. Disponível em: <https://periodicos.ufmg.br/index.php/pci/article/view/23384>.
- RIBEIRO, Mariana Ambrósio; SILVA, Márcio Bezerra da.** *Sistemas de automação de bibliotecas: um estudo investigativo-literário*. ConCI: Convergências em Ciência da Informação, Aracaju, 2019. Disponível em: <https://periodicos.ufs.br/conci/article/view/11273>.
- NASCIMENTO, Fabiana Lopes do; NASCIMENTO, Geysa Flávia Câmara de Lima.** *O sistema de gestão de biblioteca KOHA, na biblioteca do Instituto Federal da Paraíba – Campus Patos: descortinando caminhos*. SEMINÁRIO NACIONAL DE BIBLIOTECAS UNIVERSITÁRIAS (SNBU), 22., 2023, João Pessoa. *Anais* [...]. João Pessoa: FEBAB, 2023. Disponível em: <https://portal.febab.org.br/snbu2023/article/view/2845>.
- FIORILLI SOFTWARE.** *SBB – Sistema de Biblioteca*. s.d. Disponível em: <https://fiorilli.com.br/servicos/sbb-sistema-de-biblioteca/>.
- Simples Inovação.** *Mobile: o que é?* São Paulo: Simples Inovação, 2024. Disponível em: <https://simplesinovacao.com/mobile-o-que-e/>.
- ONU Brasil.** *Apenas 9% das bibliotecas brasileiras têm um site próprio, revela pesquisa*. Nações Unidas Brasil, 2019. Disponível em: <https://brasil.un.org/pt-br/79968-apenas-9-das-bibliotecas-brasileiras-t%C3%AAm-um-site-pr%C3%B3prio-revela-pesquisa>.

KOHA COMMUNITY. *History.* s.d. Disponível em: <https://koha-community.org/about/history/>.

SOPHIA. *Sistema Sophia: conheça todos os benefícios de um software de gestão.* 2023. Disponível em: <https://sophia.com.br/sistema-sophia-conheca-todos-os-beneficios-de-um-software-de-gestao>.

SOPHIA. *Sobre a Sophia.* s.d. Disponível em: [https://sophia.com.br/sobre-a-sophia/](https://sophia.com.br/sobre-a-sophia).

SOPHIA. *SophiA na Unicamp: software auxilia na integração e gestão das bibliotecas.* Sophia, 2025. Disponível em: [https://sophia.com.br/unicamp-universidade-estadual-de-campinas/](https://sophia.com.br/unicamp-universidade-estadual-de-campinas).

TANCA, Automação Comercial. *Linha TS-2000 Touch Screen da Tanca.* Disponível em: <https://www.tanca.com.br/automacao-comercial-produto.php?cod=170>.

TANCA, Automação Comercial. *Linha DT-900 Touch Screen da Tanca.* Disponível em: <https://tanca.com.br/automacao-comercial-produto.php?cod=48>.

PERGAMUM - Soluções Acadêmicas e Institucionais. Disponível em: <https://www.pergamum.pucpr.br/produtos/pergamum>.

ALURA. *HTML: o que é, a importância para a Web, como aprender e um Guia para iniciantes.* Disponível em: <https://www.alura.com.br/artigos/html?srsltid=AfmBOorYeab8RnHinmLXEJNr0el59qlyBn7rMaHWyU8Uo0BYaQUrwo7u>.

ALURA. *HTML, CSS e Javascript, quais as diferenças?* Disponível em: <https://www.alura.com.br/artigos/html-css-e-js-definicoes?srsltid=AfmBOorBNpN29eUa4DinApduLGGzkSjReeHHfzYnEIF8JLiAWV05Lqr>.

HABBEMA, Hugo. MEDIUM. *O que é esse tal de Python?* Disponível em: <https://medium.com/@habbema/o-que-%C3%A9-esse-tal-de-python-1dae07d9fad4>.

JR. COMUNICAÇÃO VISUAL. O que é: Canva. Disponível em: <https://jrbauru.com.br/glossario/o-que-e-canva-ferramenta-design-grafico-online/>.

ALURA. **VILLAIN, Mateus.** *Figma: criando protótipos interativos.* Disponível em: <https://www.alura.com.br/conteudo/figma-criando-prototipos-interativos>.

BLOG DNC. *MySQL workbench: entenda o que é e como funciona - Escola DNC.* Disponível em: <https://www.escoladnc.com.br/blog/introducao-ao-mysql-workbench>.

ALURA. **DIVINO, Bruno.** *Flask: crie uma webapp com Python.* Disponível em: <https://www.alura.com.br/conteudo/flask-crie-webapp-python>.

ALURA. *Bootstrap: o que é e como criar sites responsivos de forma rápida?* Disponível em: https://www.alura.com.br/artigos/bootstrap?srsltid=AfmBOopTQe_R59NS0At7gRYw-drQtSlh8dzISX5IfpCtsOGvU1RbIVym.

DRAW.IO. *About draw.io.* Disponível em: <https://www.drawio.com/about>

AWS. *What are AI Agents? - Agents in Artificial Intelligence Explained - AWS.* Disponível em: <https://aws.amazon.com/what-is/ai-agents/>.

ALURA. *Flutter: o que é e tudo sobre o framework.* Disponível em: <https://www.alura.com.br/artigos/flutter?srsltid=AfmBOopN8Niv0nlmwMfocRD1ZulUiPChxKrosRj55TyuM8wif7bILUWx>.

FLET. *Introduction.* Disponível em: <https://flet.dev/docs/>.

ALURA. **LOSNAK, Giulia.** *O que é metodologia ágil?* Disponível em: <https://www.alura.com.br/artigos/o-que-e-metodologia-agil?srsltid=AfmBOopvTY3wq6QvtFrd0kBfGkRAR9YKlaPHg4s7JcgDX8w1dI29qzg2>.

AWS. *O que é o Scrum? | Explicação sobre a metodologia Scrum | AWS.* Disponível em: <https://aws.amazon.com/pt/what-is/scrum/>.

FACEBOOK; ACERT. *Técnicas de Elicitação de Requisitos» ACerT - Academia Brasileira de Certificações e Treinamentos.* Disponível em: <https://acertbr.com.br/tecnicas-de-elicitacao-de-requisitos/>.

DEVMEDIA. *Técnicas para levantamento de Requisitos.* Disponível em: <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>.

SULIVAM. *Levantamento de Requisitos: Importância e Custos.* Disponível em: <https://sulivam.com.br/levantamento-de-requisitos-importancia-custos-como-fazer/>.

CEDRO. PRECioso, Vitor. *Como realizar o Levantamento de Requisitos no desenvolvimento de software.* Disponível em: <https://www.cedrotech.com/blog/levantamento-de-requisitos-e-desenvolvimento-de-softwares/>.

JAIN, Anushtha. VISURE SOLUTIONS, 2025. *O que são requisitos não funcionais: tipos, exemplos e abordagens.* Disponível em: <https://visuresolutions.com/pt/alm-guide/non-functional-requirements/>.

JAIN, Anushtha. VISURE SOLUTIONS, 2025. *O que são requisitos funcionais: exemplos e modelos.* Disponível em: <https://visuresolutions.com/pt/alm-guide/functional-requirements/>.

PUCPR – Pontifícia Universidade Católica do Paraná. *Tecnologia da Informação e Comunicação (TIC).* Disponível em: <https://www.pucpr.br/areas-estrategicas/tic/>.

LOUISE, Maiza. MEDIUM, 2023. *O que é Flutter?* Disponível em: <https://medium.com/@maizalouise/o-que-%C3%A9-flutter-d0848268287c>.

ARAÚJO, Raquel. %*Como Criar Apps Flutter Usando Python e Flet? Descubra já!%.* Disponível em: <https://www.hashtagtreinamentos.com/como-criar-apps-flutter-com-python>.

BIBLIOTECA VIRTUAL. *Manual de Uso Plataforma do Aluno.* Disponível em: https://staticbv.bvirtual.com.br/conteudos/84/Manual-de-uso_Usuario_v5.pdf.

LUCIDCHART. *Diagrama de caso de uso UML: O que é, como fazer e exemplos.* Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>.

LUCIDCHART. *O que é um diagrama entidade relacionamento?* Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento>.

ESTRATÉGIA. FERREIRA, Cristiane. *Engenharia de Software em Destaque: Levantamento de Requisitos.* Disponível em:

<https://www.estrategiaconcursos.com.br/blog/engenharia-software-levantamento-requisitos/>.

MIRO. *O que é Brainstorming? Como fazer? Técnicas e modelos editáveis.* Disponível em: <https://miro.com/pt/brainstorming/o-que-e-brainstorming/>.