



GreenBook - Software de gerenciamento de livros

Gabriel Alves Fortunato - 820142037

Gustavo Miguel de Sá Pito Moreno – 820132372

Lucas Muniz Nocera – 820132951

SUMÁRIO

1 INTRODUÇÃO	3
2 PROJETO	3
2.1 Definição do Projeto	3
2.2 Objetivos	4
3 MODELAGEM	5
3.1 Caso de uso	6
3.2 Requisitos funcionais e não funcionais	7
3.3 Modelo conceitual do banco de dados	9
3.4 Modelo lógico do banco de dados	9
3.5 Diagrama de classes	10
4 DESENVOLVIMENTO	10
4.1 Apresentação do código	10
4.2 Banco de dados	11
5 CONCLUSÃO	13

1 Introdução

No âmbito do 8º semestre do curso de engenharia de computação, fomos desafiados a aplicar e aprimorar nossos conhecimentos técnicos ao desenvolver um sistema utilizando a linguagem Java. O escopo do projeto demandou a implementação de uma solução interativa e eficiente, incorporando uma conexão com um banco de dados. Dessa forma, foi proposta a criação de uma interface de acesso a uma coleção de livros, proporcionando ao usuário a capacidade de gerenciar, avaliar e favoritar obras conforme suas preferências e necessidades.

O sistema visa não apenas demonstrar nossa proficiência na linguagem Java, mas também destacar nossa habilidade em projetar e implementar uma arquitetura eficiente de banco de dados. A proposta central do projeto consiste em fornecer aos usuários uma experiência intuitiva e personalizada, possibilitando a exploração de um acervo diversificado de livros.

Ao longo deste documento, discutiremos os desafios enfrentados durante o desenvolvimento, as escolhas de design que nortearam nossas decisões e as soluções implementadas para atender aos requisitos funcionais e não funcionais do sistema.

2 PROJETO

2.1 Definição do Projeto

Ao desenvolver o sistema, demos atenção ao seu design, buscando não apenas eficiência técnica, mas também uma abordagem consciente em relação ao meio ambiente. O nome escolhido para o sistema, "GreenBook", reflete essa preocupação ambiental. Emergindo como uma alternativa eco-friendly, incentivando a leitura sem comprometer os recursos naturais.

No processo de design da interface do GreenBook, escolhemos as cores verde e branco como elementos visuais predominantes. O verde, simbolizando a natureza, transmite a mensagem da consciência. O branco, por sua vez, representa a

simplicidade e a clareza de propósito, garantindo uma experiência de usuário fluida e intuitiva.

Acreditamos que a experiência do usuário vai além da funcionalidade, estendendo-se à estética e à sensação que o sistema transmite. A paleta de cores selecionada não apenas cria uma interface visualmente atraente, minimalista e moderna, mas também reforça a mensagem ecológica que o GreenBook busca transmitir.

Sobre a funcionalidade do sistema, nos preocupamos em programar um software simples e eficiente, onde os usuários possam visualizar, gerenciar (adicionar, editar e excluir), avaliar e favoritar todos os livros inseridos no sistema.

2.2 Objetivos

Nosso principal objetivo foi absorver integralmente o processo de desenvolvimento do sistema, indo além da codificação e mergulhando em todas as fases do ciclo de vida do software. A proposta apresentada para o GreenBook buscou não apenas criar um sistema funcional, mas também destacar a importância de uma abordagem abrangente e integrada.

A jornada iniciou-se com a fase de modelagem, onde aplicamos diversas técnicas para conceber uma representação abstrata e compreensível do sistema. Diagramas de classes definiram a estrutura do GreenBook, destacando as relações entre os diversos elementos. Os requisitos foram identificados, e modelos lógicos foram desenvolvidos para criar uma base sólida para a implementação.

A transição para a etapa de design envolveu a criação de diagramas de atividade e casos de uso, proporcionando uma visão detalhada das interações entre os usuários e o sistema. Esse processo de planejamento não apenas facilitou o desenvolvimento subsequente, mas também contribuiu para a criação de um sistema centrado no usuário.

A programação foi realizada em conformidade com as práticas modernas e padrões de codificação, garantindo funcionalidade, manutenibilidade e escalabilidade. A escolha da

linguagem Java proporcionou uma base sólida para a implementação eficiente das funcionalidades do GreenBook.

A conexão com o banco de dados representou um desafio significativo, mas também uma oportunidade de aprimorar nossas habilidades em gerenciamento de dados. A estrutura do banco de dados foi planejada para garantir uma manipulação eficiente das informações, proporcionando uma base para o armazenamento e recuperação de dados.

Em todo esse processo, mantivemos uma atenção especial à experiência do usuário, visando um sistema usual e fácil de lidar. A interface foi projetada para ser intuitiva, permitindo que os usuários explorem e interajam com a plataforma de forma descomplicada.

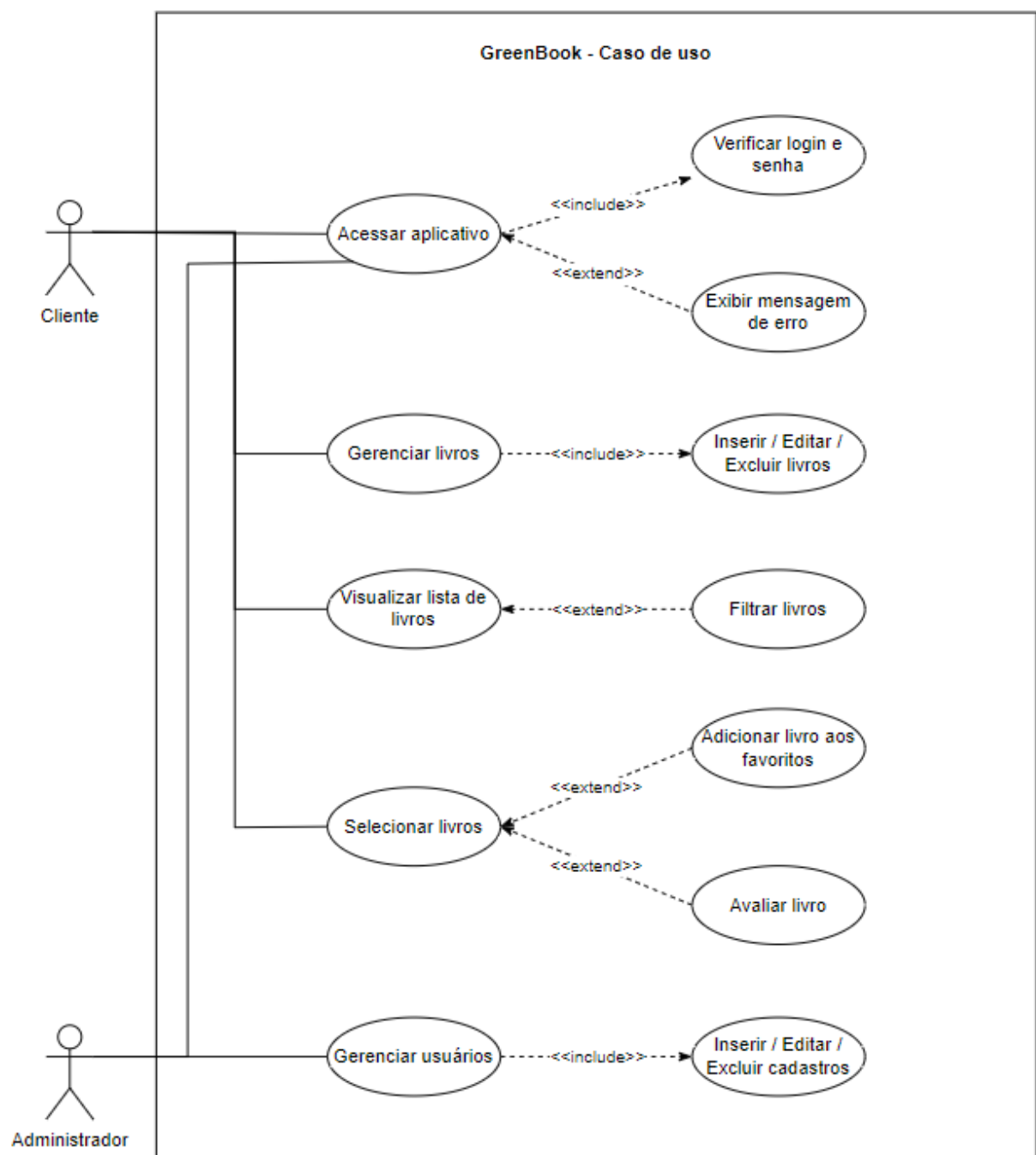
Ao final deste projeto, consolidamos nossos conhecimentos em programação, desenvolvimento e internalizamos a importância de uma abordagem holística, desde a concepção até a entrega de uma solução completa e eficaz.

3 Modelagem

A modelagem foi um processo crucial no desenvolvimento do sistema, pois desempenhou um papel fundamental na concepção, visualização, especificação e documentação de um sistema antes de sua implementação.

Durante a modelagem do software foi desenvolvido o diagrama de caso de uso, os requisitos funcionais e não funcionais, o diagrama de classes, o modelo conceitual e lógico do banco de dados, os diagramas de sequências e os diagramas de atividades do sistema.

3.1 Caso de uso



3.2 Requisitos funcionais e não funcionais

Requisitos Funcionais

Acesso de Usuários: O sistema deve permitir o acesso de usuários já cadastrados através de verificação de login.

Cadastro de Usuários: O sistema deve permitir o cadastro de novos usuários somente pelo administrador, incluindo informações como nome de usuário, senha, nome completo, sexo, idade, e-mail e endereço.

Edição de Usuários: O sistema deve permitir a edição de informações de usuários já existentes somente pelo administrador.

Exclusão de Usuários: O sistema deve permitir a exclusão de informações de usuários já existentes somente pelo administrador.

Visualização de Usuários: O sistema deve permitir que somente o administrador visualize as informações de todos os usuários já existentes.

Cadastro de Livros: O sistema deve permitir o cadastro de novos livros, incluindo informações como título, autor, gênero e usuário responsável pelo cadastro.

Avaliação de Livros: O sistema deve permitir que os usuários avaliem os todos os livros cadastrados.

Adicionar Livros aos Favoritos: O sistema deve permitir que os usuários adicionem livros na lista de favoritos.

Visualização de Livros: O sistema deve permitir aos usuários visualizar todos os livros e suas informações.

Atualização de Informações de Livros: O sistema deve permitir a atualização das informações dos livros.

Exclusão de Livros: O sistema deve permitir a exclusão de livros quando necessário.

Requisitos Não Funcionais

Desempenho: O sistema deve responder às consultas de pesquisa e atualizações de informações de livros em até 5 segundos, mesmo com grande volume de dados.

Usabilidade: A interface do usuário deve ser intuitiva e fácil de usar, com uma curva de aprendizado mínima para os usuários.

Segurança: O acesso ao sistema deve ser seguro e exigir autenticação, com proteção adequada dos dados armazenados, utilizando criptografia e políticas de privacidade.

Disponibilidade: O sistema deve estar disponível 99% do tempo, excluindo manutenções planejadas.

Compatibilidade: O sistema deve ser compatível com dispositivos desktops, garantindo uma experiência consistente. Também deverá ser compatível com o sistema operacional Windows, e com o gerenciador de banco de dados MySQL.

Escalabilidade: O sistema deve ser escalável para lidar com um aumento substancial no número de usuários e livros, sem degradação significativa do desempenho.

Manutenibilidade: O sistema deve ser projetado de forma modular e bem documentado para facilitar futuras atualizações, manutenção e expansões.

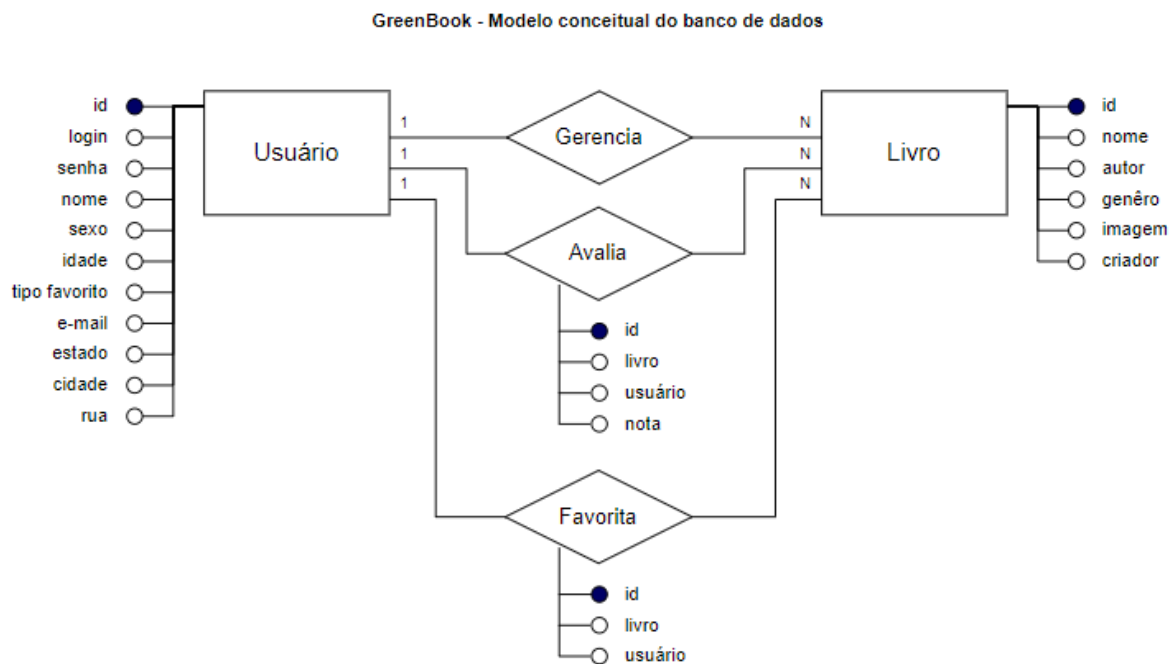
Confiabilidade: O sistema deve garantir que os dados sejam armazenados de maneira confiável e não sejam perdidos devido a falhas no sistema.

Tempo de Resposta: O sistema deve responder às ações dos usuários, como cliques e submissões de formulários, em menos de 300 milissegundos para garantir uma experiência responsiva.

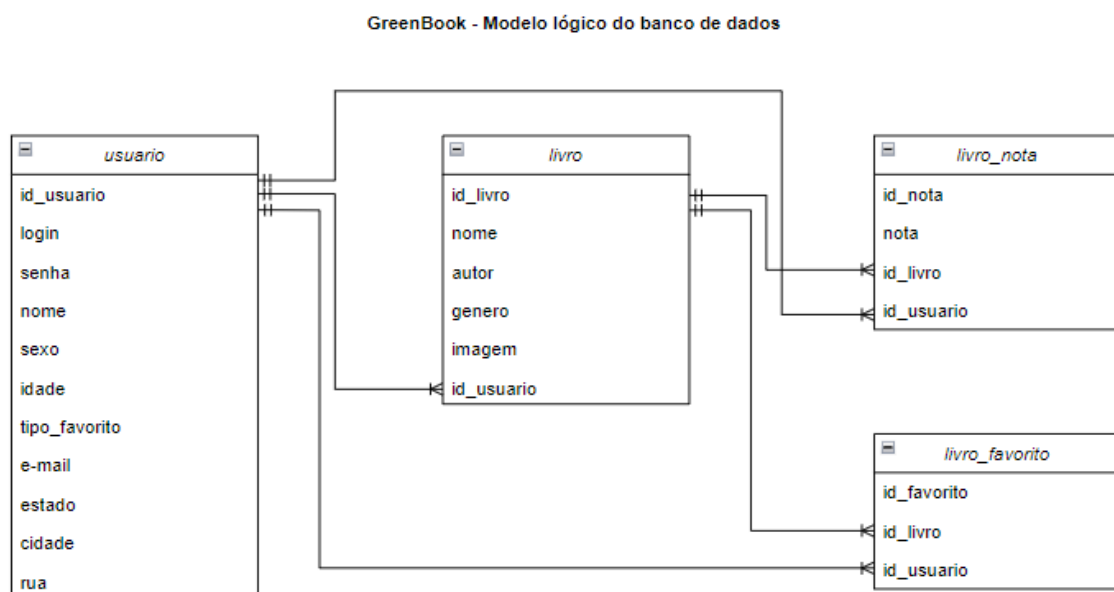
Regulamentação e Conformidade: O sistema deve estar em conformidade com as leis e regulamentações locais, especialmente em relação à proteção de dados e privacidade.

Performance em Carga Máxima: O sistema deve manter um desempenho aceitável mesmo em situações de carga máxima, garantindo a satisfação dos usuários

3.3 Modelo conceitual do banco de dados

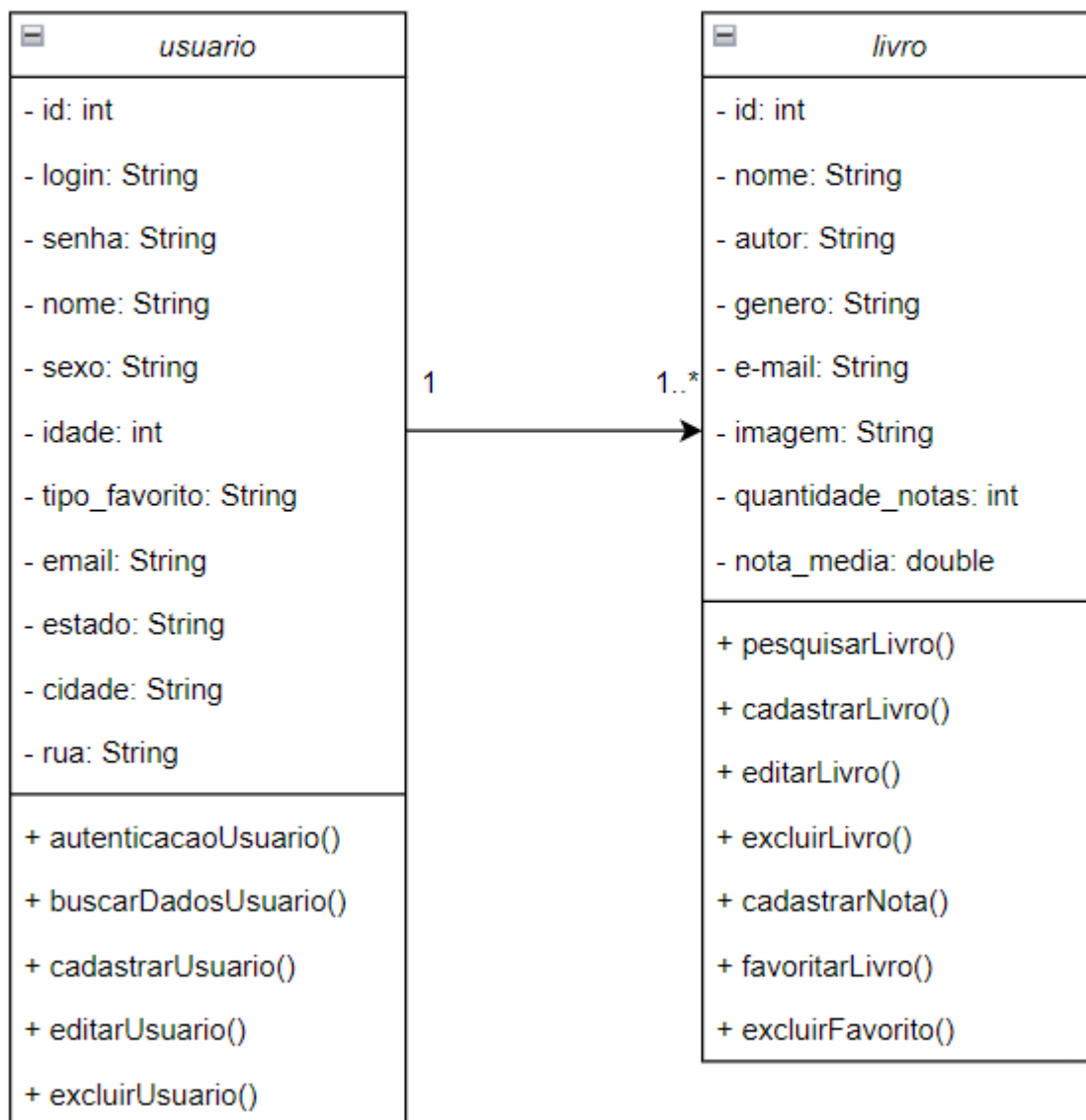


3.4 Modelo lógico do banco de dados



3.5 Diagrama de classes

GreenBook - Diagrama de classes



4 Desenvolvimento

4.1 Apresentação do código

Como já citado anteriormente, o código foi desenvolvido na linguagem Java utilizando os princípios de programação orientada à objeto. A programação de todo o código

seguiu o padrão de design Java MVC, onde o software foi dividido em 3 camadas, sendo elas: VIEW, DTO e DAO.

Todo o código responsável pelo sistema encontra-se em um repositório público do github, cujo link é: https://github.com/GustavoMiguel28/greenbook_A3.

4.2 Banco de dados

O sistema foi conectado a um banco de dados MySQL, para assim armazenar todos os dados gerados pelos usuários.

Essa conexão com o banco de dados desempenha um papel crucial no projeto GreenBook, proporcionando diversas vantagens que contribuem para a eficácia e funcionalidade do sistema. Aqui estão algumas razões para a importância da conexão com banco de dados neste projeto específico:

1. **Armazenamento Eficiente de Dados:** conexão com o banco de dados permite armazenar de maneira organizada e eficiente informações essenciais para o GreenBook, como detalhes dos livros, perfis de usuários, avaliações e preferências. Isso facilita a recuperação e manipulação dos dados conforme necessário.
2. **Persistência de Dados:** O banco de dados fornece persistência aos dados do sistema, ou seja, os dados são mantidos mesmo quando a aplicação é encerrada. Isso é fundamental para garantir que as informações não se percam entre as sessões de uso, permitindo que os usuários retornem e encontrem seus dados intactos.
3. **Facilitação da Escalabilidade:** Com um banco de dados bem projetado, o GreenBook pode lidar com um grande volume de informações sem sacrificar o desempenho. Isso é especialmente importante à medida que o sistema cresce e mais usuários interagem com a plataforma, garantindo que a aplicação permaneça responsiva e eficiente.
4. **Gerenciamento de Relacionamentos:** O uso de um banco de dados possibilita a criação de relacionamentos entre diferentes conjuntos de dados. Por exemplo, é

possível estabelecer conexões entre os usuários e os livros que eles avaliaram, facilitando consultas complexas e análises de dados.

5. Atualização Dinâmica de Conteúdo: Com a conexão com o banco de dados, é possível atualizar dinamicamente o conteúdo do GreenBook. Novos livros, avaliações ou recursos podem ser facilmente adicionados, modificados ou removidos sem a necessidade de alterações significativas no código-fonte da aplicação.

6. Segurança e Controle de Acesso: O banco de dados permite implementar medidas de segurança, controlando o acesso aos dados sensíveis. Isso é vital para proteger informações confidenciais dos usuários e garantir que apenas usuários autorizados tenham acesso a determinados recursos e funcionalidades.

7. Backup e Recuperação: A conexão com banco de dados facilita a implementação de estratégias de backup, garantindo que os dados críticos estejam protegidos contra perdas acidentais. Em caso de falhas no sistema, é possível recuperar os dados a partir dos backups.

Portanto, a conexão com banco de dados não é apenas uma parte técnica do GreenBook, mas uma componente estratégica que contribui para a robustez, flexibilidade e eficiência do sistema como um todo.

Para satisfazer as necessidades do software, o banco de dados possui 4 tabelas, sendo elas: usuario, livro, livro_nota e livro_favorito.

1. Tabela `usuario`:

- `id_usuario` (Chave Primária): Identificador único para cada usuário.
- `login_usuario`: Nome de login do usuário.
- `senha_usuario`: Senha do usuário.
- `nome_usuario`: Nome do usuário.
- `sexo_usuario`: Sexo do usuário.
- `idade_usuario`: Idade do usuário.
- `tipo_usuario`: Tipo de livro preferido do usuário.

- `email_usuario`: Endereço de e-mail do usuário.
- `estado_usuario`: Estado do usuário.
- `cidade_usuario`: Cidade do usuário.
- `rua_usuario`: Rua do usuário.

2. Tabela `livro`:

- `id_livro` (Chave Primária): Identificador único para cada livro.
- `nome_livro`: Nome do livro.
- `autor_livro`: Autor do livro.
- `genero_livro`: Gênero do livro.
- `imagem_livro`: Caminho para a imagem de capa do livro.
- `id_usuario` (Chave Estrangeira): Referência ao usuário que cadastrou o livro.

3. Tabela `livro_nota`:

- `id_nota` (Chave Primária): Identificador único para cada nota.
- `id_livro` (Chave Estrangeira): Referência ao livro que recebeu a nota.
- `id_usuario` (Chave Estrangeira): Referência ao usuário que deu a nota.
- `nota_livro`: Nota atribuída ao livro pelo usuário.

4. Tabela `livro_favorito`:

- `id_favorito` (Chave Primária): Identificador único para cada entrada na tabela.
- `id_usuario` (Chave Estrangeira): Referência ao usuário que marcou o livro como favorito.
- `id_livro` (Chave Estrangeira): Referência ao livro marcado como favorito.

5 Conclusão

Ao longo deste projeto, exploramos o desenvolvimento do GreenBook, realizado durante o semestre de engenharia de computação. Desde a sua concepção até a implementação prática, o GreenBook representou mais do que a criação de um sistema

eficiente; foi uma jornada que incorporou os princípios fundamentais do desenvolvimento de software.

O projeto começou com a visão de integrar completamente o processo de desenvolvimento, abrangendo desde as fases iniciais de modelagem até a programação efetiva, conectando-se a um banco de dados. A escolha de não utilizar livros físicos, refletida no nome "GreenBook", ressalta a conscientização ambiental e a busca por alternativas sustentáveis.

A modelagem prévia, representada por diagramas de classes, requisitos, modelos lógicos, diagramas de atividade e casos de uso, estabeleceu uma base sólida para a programação posterior. A linguagem Java foi escolhida não apenas pela sua eficiência, mas também pela sua flexibilidade e suporte robusto à orientação a objetos.

A conexão com o banco de dados surgiu como um elemento essencial, proporcionando armazenamento eficiente, persistência de dados, escalabilidade, gerenciamento de relacionamentos e controle de acesso.

A estética da interface, refletida nas escolhas de cores verde e branco, foi cuidadosamente planejada, adicionando uma camada de usabilidade e intuitividade ao sistema.

