**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Tiago Daniel de Sousa Filho**

**BiblioTech**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

Uma biblioteca tradicional enfrenta desafios na administração dos empréstimos, com registros em papel, atrasos frequentes nas devoluções e dificuldades para acompanhar o status dos livros. Para melhorar a eficiência e organização, os funcionários decidiram implementar um sistema gerenciador para a equipe.

Com esse novo sistema, os bibliotecários podem rapidamente acessar informações sobre o acervo, verificar quais livros estão disponíveis ou emprestados e registrar novas retiradas e devoluções de forma digital.

Os funcionários também têm a capacidade de enviar lembretes de devolução diretamente para os leitores e gerenciar solicitações de renovação, facilitando a comunicação e reduzindo problemas com atrasos. A automação dessas tarefas burocráticas permite que a equipe se concentre em projetos educativos e eventos culturais, como oficinas de leitura e exposições temáticas.

Esse sistema interno trouxe mais agilidade e precisão ao trabalho dos bibliotecários, tornando o gerenciamento do acervo mais simples e eficiente. Agora, a equipe podia oferecer um serviço mais ágil e organizado, fortalecendo o papel da biblioteca como um espaço acolhedor de aprendizado e cultura para toda a comunidade.

**2 REQUISITOS DO SISTEMA**

Os requisitos de sistema são fundamentais para definir expectativas, orientar o desenvolvimento e garantir que o produto atenda às necessidades dos usuários, minimizando retrabalhos e custos.

**2.1 Requisitos funcionais**

Requisitos funcionais descrevem as funcionalidades que um sistema deve ter. Eles são essenciais para garantir que o software atenda às necessidades dos usuários e suporte processos específicos, orientando o desenvolvimento. A tabela 1 a seguir representa os requisitos funcionais do BiblioTech.

|  |  |
| --- | --- |
| Requisito Funcional | |
| RF001 | Cadastrar Livro |
| RF002 | Editar Livro |
| RF003 | Excluir Livro |
| RF004 | Cadastrar Aluno |
| RF005 | Editar Aluno |
| RF006 | Excluir Aluno |
| RF007 | Empréstimo Livro |
| RF008 | Registrar Devolução |
| RF009 | Consultar Empréstimo |

Tabela 1 Requisitos Funcionais

**2.2.1 RF001 – Cadastrar Livro**

Descrição: O sistema deve permitir que o usuário cadastre novos livros no acervo da biblioteca.

Prioridade: Alta

Entrada:

* Título do livro (obrigatório).
* Autor(es) (obrigatório).
* Editora (obrigatório).
* Ano de Publicação
* ISBN
* Quantidade Total (obrigatório).
* Quantidade Disponível (obrigatório).
* Valor Aquisição
* Status do Livro Emprestado

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao cadastrar o livro.
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro.

Pós-condição:

* O livro cadastrado deve ser exibido na lista de acervo disponível.

**2.2.2 RF002 – Editar Livro**

Descrição: O sistema deve permitir que o usuário edite as informações de um livro previamente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do livro a ser editado (ID ou ISBN).
* Título do livro a ser editado.
* Autor do livro a ser editado.
* Editora do livro a ser editado.
* Ano de Publicação do livro a ser editado.
* Quantidade Total do livro a ser editado.
* Quantidade Disponível do livro a ser editado.
* Valor de Aquisição do livro a ser editado.
* Status de Empréstimo do livro a ser editado.

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja duplicidade ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas na lista de acervo e nos registros de empréstimos associados ao livro.

**2.2.3 RF003 – Excluir Livro**

Descrição: O sistema deve permitir que o usuário exclua um livro do acervo.

Prioridade: Média

Entrada:

* Identificação do livro a ser excluído (ID).

Processamento:

* O sistema deve verificar se há empréstimos ativos ou pendentes associados ao livro.
* Se houver empréstimos associados, o sistema deve impedir a exclusão e informar o usuário.

Saída:

* Mensagem de sucesso ao excluir o livro.
* Mensagem de erro caso existam empréstimos associados impedindo a exclusão.

Pós-condição:

* O livro deve ser removido da lista de acervo e não deve aparecer nas buscas e consultas.

**2.2.4 RF004 – Cadastrar Aluno**

Descrição: O sistema deve permitir que o usuário cadastre novos alunos no acervo da biblioteca.

Prioridade: Alta

Entrada:

* RA (obrigatório).
* Nome (obrigatório).
* Sobrenome (obrigatório).
* Data de Nascimento
* Endereço
* E-mail
* Celular (obrigatório).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao cadastrar o aluno.
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro.

Pós-condição:

* O aluno cadastrado deve ser exibido na lista de frequentadores da biblioteca.

**2.2.5 RF005 – Editar Aluno**

Descrição: O sistema deve permitir que o usuário edite as informações de um aluno previamente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do aluno a ser editado (ID).
* Nome do Aluno a ser editado.
* Sobrenome do Aluno a ser editado.
* Data de Nascimento do Aluno a ser editado.
* Endereço Aluno a ser editado.
* Email do Aluno a ser editado.
* Celular do Aluno a ser editado.

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja duplicidade ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas na lista de frequentadores e nos registros de empréstimos associados ao aluno.

**2.2.6 RF006 – Excluir Livro**

Descrição: O sistema deve permitir que o usuário exclua um aluno da lista de frequentadores.

Prioridade: Média

Entrada:

* Identificação do aluno a ser excluído (ID).

Processamento:

* O sistema deve verificar se há devoluções pendentes associadas ao aluno.
* Se houver devoluções pendentes associadas, o sistema deve impedir a exclusão e informar o usuário.

Saída:

* Mensagem de sucesso ao excluir o aluno.
* Mensagem de erro caso existam devoluções pendentes associadas impedindo a exclusão.

Pós-condição:

* O aluno deve ser removido da lista de frequentadores e não deve aparecer nas buscas e consultas.

**2.2.7 RF007 – Empréstimo de Livro**

Descrição: O sistema deve permitir que o usuário registre o empréstimo de um ou mais livros para um aluno cadastrado.

Prioridade: Alta

Entrada:

* Identificação do aluno (número de matrícula).
* Identificação do(s) livro(s) a ser(em) emprestado(s) (ID do livro).
* Data de início do empréstimo (obrigatória).
* Data prevista de devolução.

Processamento:

* O sistema deve verificar a disponibilidade dos livros no acervo.
* O sistema deve reduzir a quantidade disponível do livro emprestado.
* O sistema deve associar o empréstimo ao aluno e ao(s) livro(s) selecionado(s).

Saída:

* Mensagem de sucesso ao registrar o empréstimo.
* Mensagem de erro caso algum dos livros não esteja disponível.

Pós-condição:

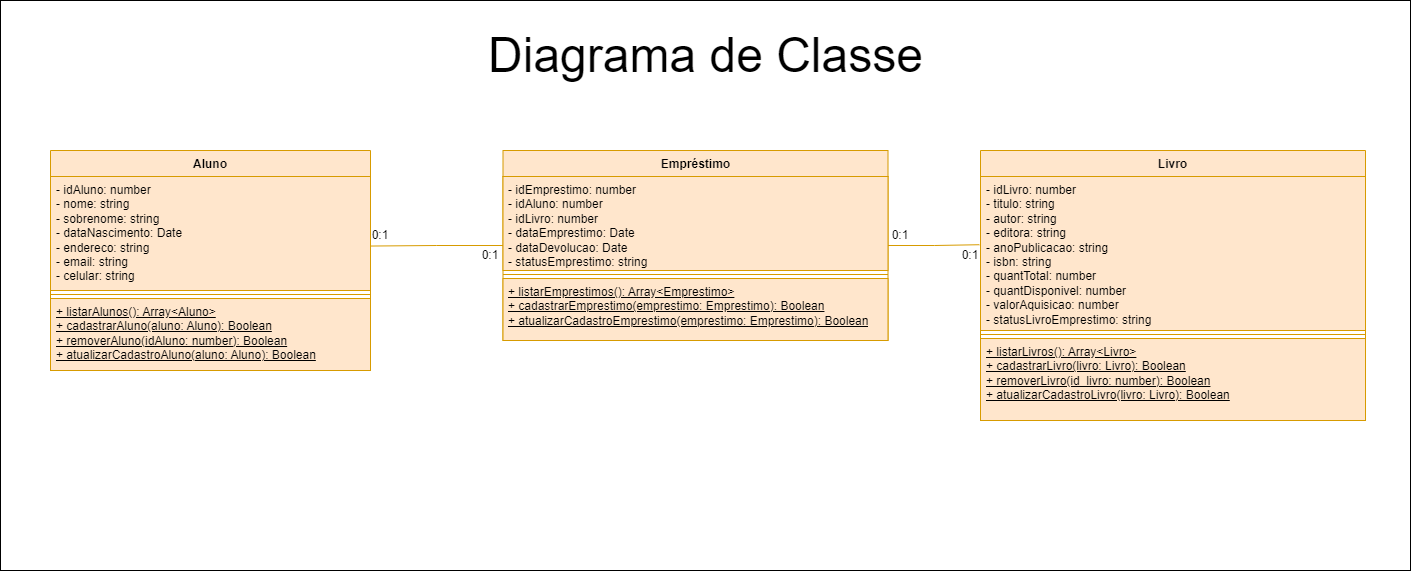
* O empréstimo deve ser registrado com status "Ativo".
* A quantidade disponível do livro deve ser atualizada no sistema.

**3 DIAGRAMAS DO SISTEMA**

Os diagramas desempenham um papel crucial no trabalho dos programadores, sendo fundamentais para o planejamento, desenvolvimento e manutenção de sistemas. Eles oferecem uma representação visual clara e detalhada dos componentes do sistema, facilitando a compreensão de como as diferentes partes interagem entre si e com o ambiente externo. Isso permite que os desenvolvedores visualizem a arquitetura, os fluxos de dados e os processos de forma mais intuitiva, auxiliando na identificação de potenciais problemas e na implementação de soluções mais eficazes. Além de minimizar erros durante a codificação, os diagramas contribuem para a documentação do projeto, melhoram a comunicação entre a equipe e stakeholders, e tornam o processo de planejamento e desenvolvimento mais eficiente e organizado.

**3.1 Diagrama de Classe**

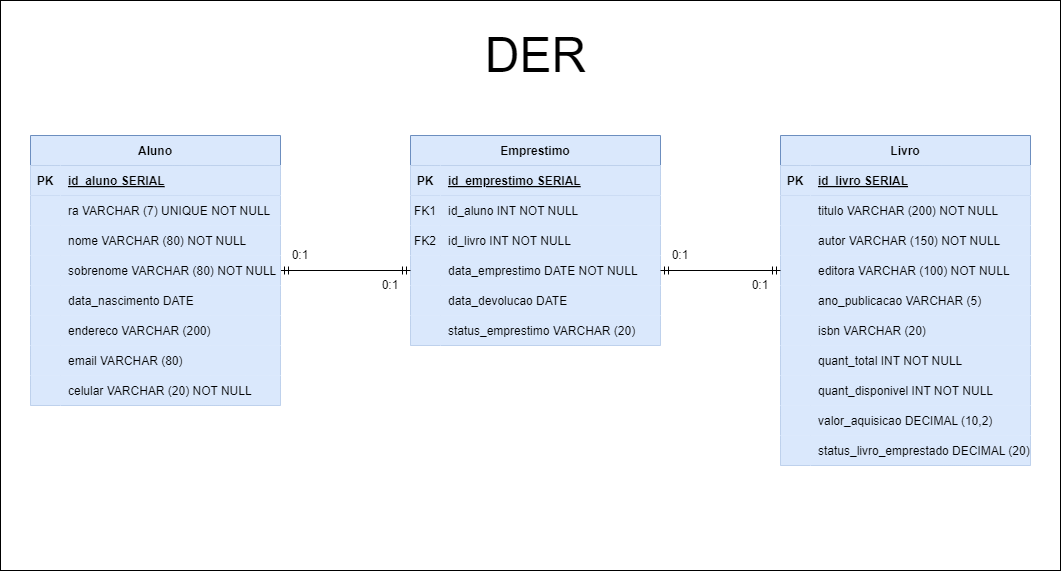
O diagrama de classes é uma ferramenta valiosa para programadores backend, pois oferece uma representação visual das classes e suas relações dentro do sistema. Ele ajuda a modelar a estrutura de dados, permitindo que os desenvolvedores entendam como os objetos interagem entre si. Além disso, o diagrama facilita a identificação de atributos e métodos, contribuindo para uma arquitetura mais organizada e eficiente. A imagem 1 a seguir representa o diagrama de classe do BiblioTech.



**3.2 Banco de Dados**

No BiblioTech, utilizamos o pgAdmin, que é uma ferramenta de gerenciamento de banco de dados para o PostgreSQL, oferecendo uma interface gráfica intuitiva que facilita a administração de bancos de dados. Ele permite a criação, edição e exclusão de objetos como tabelas, esquemas e índices, além de fornecer suporte a consultas SQL com recursos de auto-completar e visualização de resultados. A interface é organizada em painéis que exibem informações sobre objetos do banco de dados, tornando a navegação e o gerenciamento mais simples e eficiente.

O Diagrama Entidade-Relacionamento (DER) é uma representação visual que ilustra as entidades de um sistema e suas relações. Ele é fundamental na fase de modelagem de dados, pois ajuda a identificar e organizar a estrutura do banco de dados, garantindo que todos os requisitos sejam atendidos antes da implementação. Isso facilita a comunicação entre a equipe de desenvolvimento e os stakeholders, minimizando erros e melhorando a eficiência do projeto. A imagem a seguir representa o DER do BiblioTech.



**4 ROTAS DA APLICAÇÃO – BACK-END**

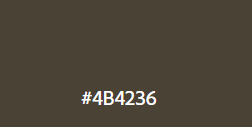
As rotas são responsáveis por definir os caminhos de acesso dentro de uma aplicação, conectando solicitações feitas pelos usuários às funcionalidades ou dados apropriados. Elas estabelecem qual recurso será acionado com base na URL e no método HTTP (como GET, POST, PUT ou DELETE), organizando a comunicação entre o cliente e o servidor. Além disso, as rotas podem incluir parâmetros e validações para garantir que apenas requisições válidas sejam processadas, contribuindo para a estruturação e segurança do sistema.

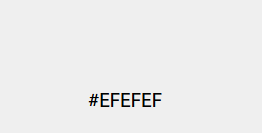
|  |  |  |
| --- | --- | --- |
| **Função** | **Método** | **Rota** |
| Cadastrar Aluno | POST | /novo/aluno |
| Cadastrar Livro | POST | /novo/livro |
| Listar Aluno | GET | /lista/alunos |
| Listar Livro | GET | /lista/livros |
| Listar Empréstimo | GET | /lista/emprestimos |

5 INTERFACE E EXPERIÊNCIA DO USUÁRIO

Uma **interface** é um contrato que define um conjunto de métodos ou propriedades que uma classe deve implementar. Seu objetivo é garantir que diferentes classes sigam o mesmo padrão, promovendo consistência, flexibilidade e reutilização de código. Ela permite que diferentes objetos sejam tratados de forma uniforme, independentemente de suas implementações específicas.

5.2 Paleta de Cores

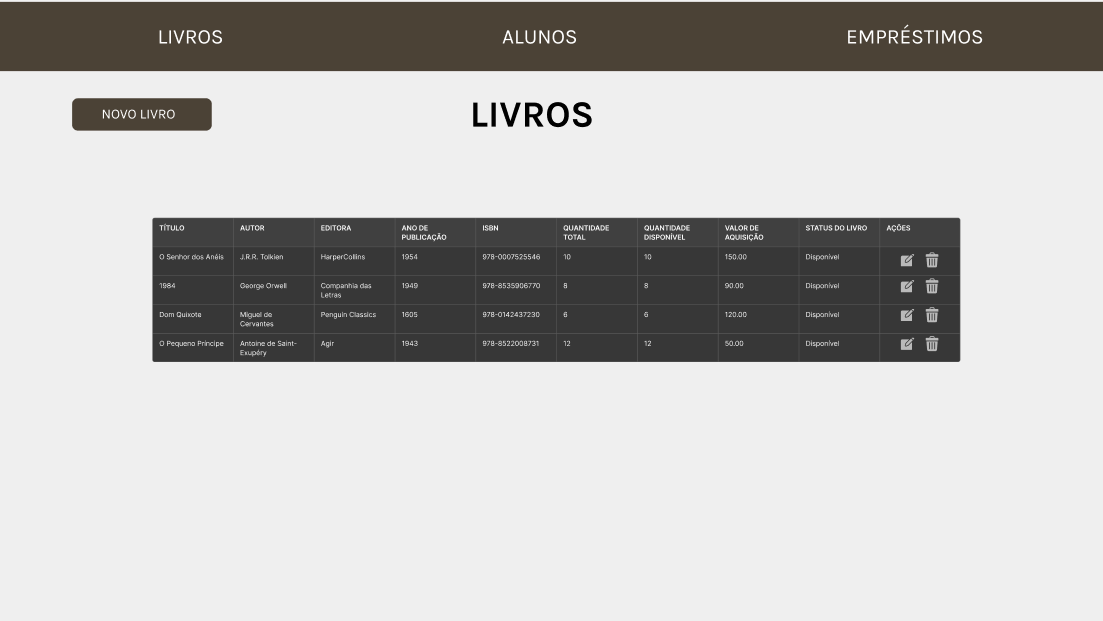
Eu decidi utilizar um Marrom mais escuro para ter um contraste com as letras brancas e o Cinza mais claro para ter um site visualmente confortável aos olhos. 



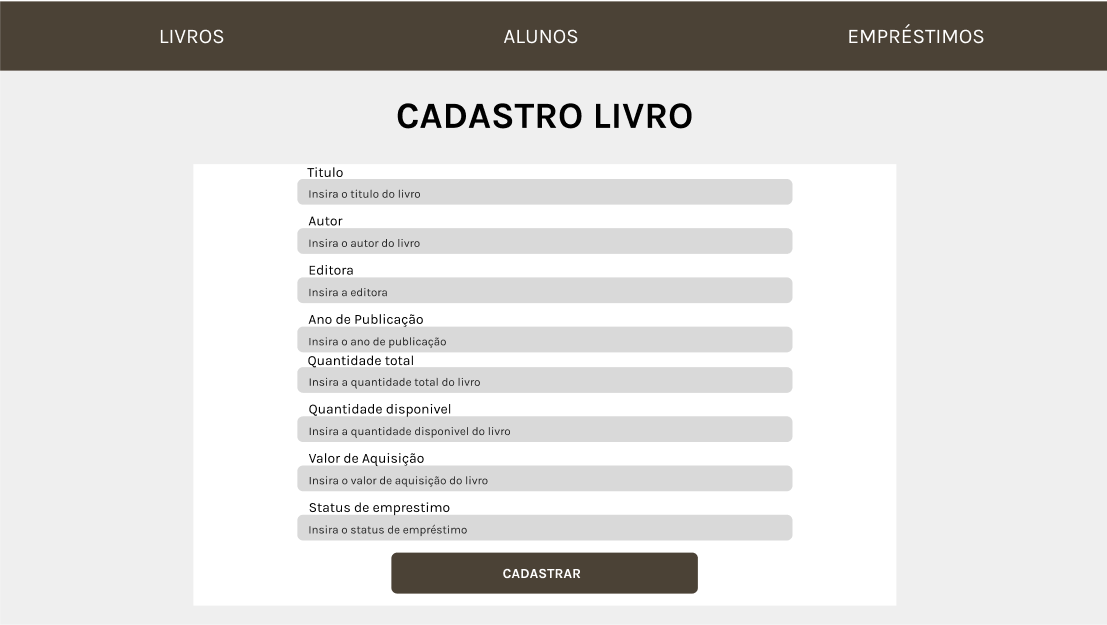
5.3 Mockup



Página 1 – Página Inicial



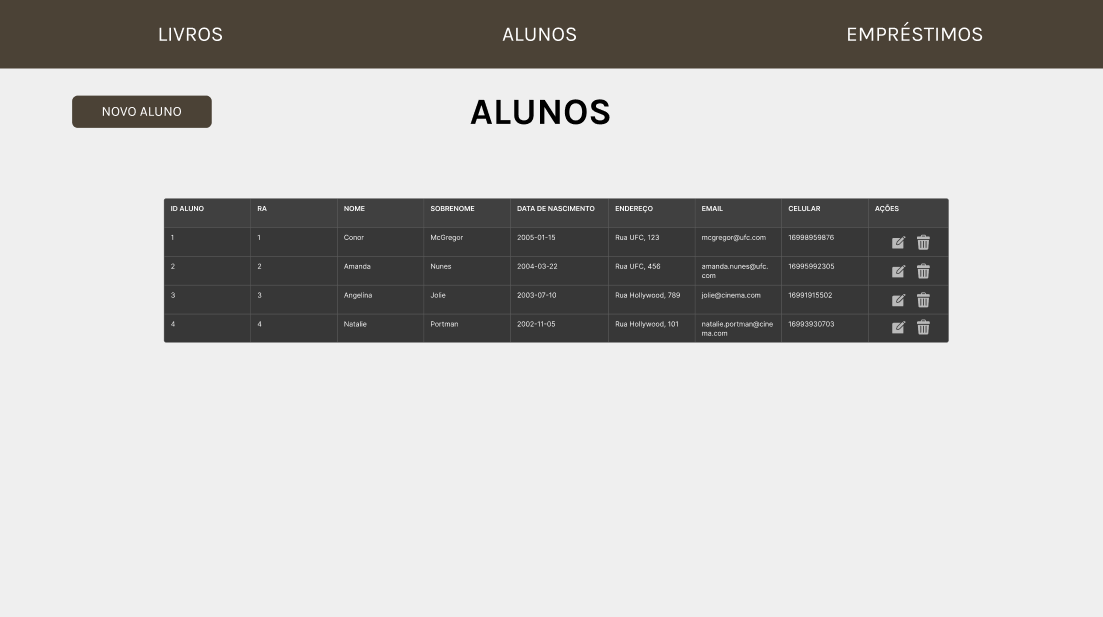
Página 2 – Página de Listagem dos Livros



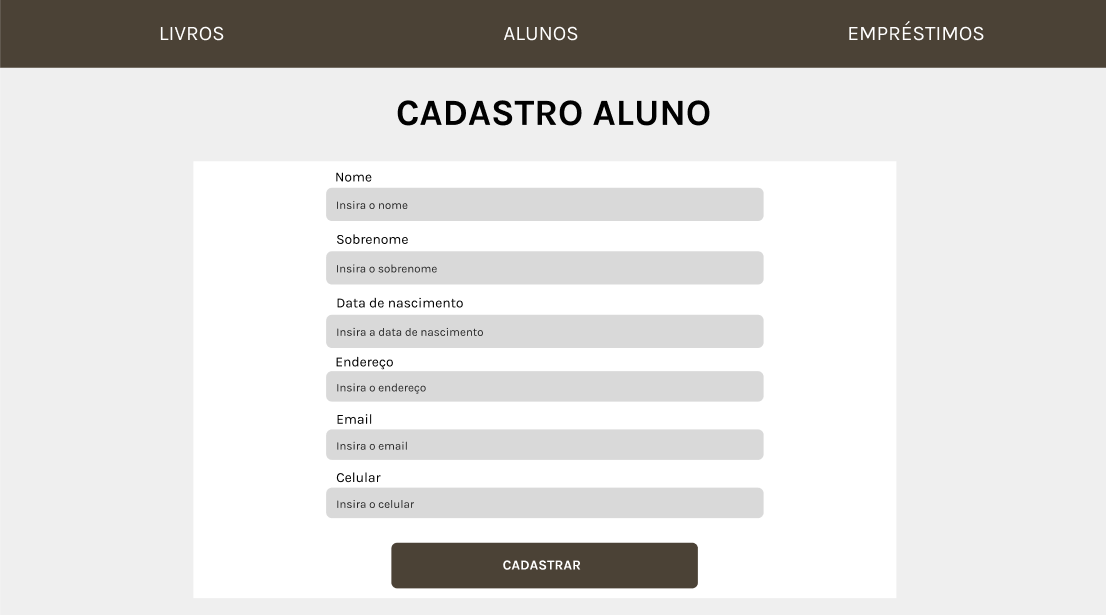
Página 3 – Página Cadastro de Livro



Página 4 – Página de Atualização de Livro



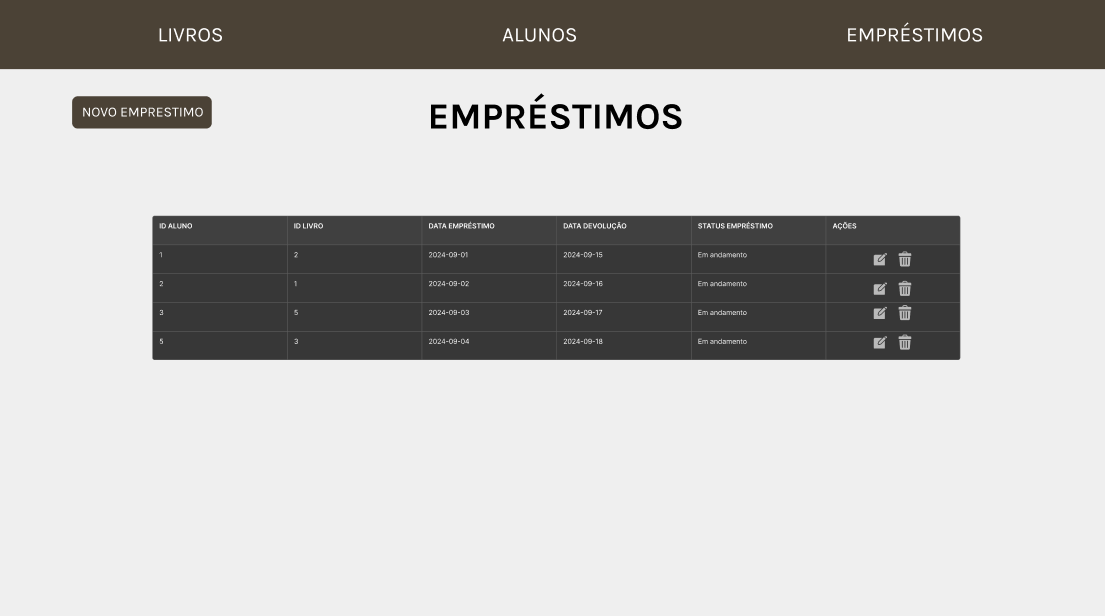
Página 5 – Página de Listagem dos Alunos



Página 6 – Página de Cadastro de Aluno



Página 7 – Página atualização de Aluno



Página 8 – Página de Listagem dos Empréstimos



Página 9 – Página de Cadastro de Empréstimo



Página 10 – Página de Atualização de Empréstimo

BIBLIOGRAFIA

Link das documentações utilizadas e livros consultados