**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**João Pedro Temponi de Jesus**

**BiblioManager**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

Uma biblioteca escolar é um espaço onde os alunos têm a oportunidade de explorar ideias, ampliar seus horizontes e mergulhar em mundos de conhecimento. No entanto, gerenciar esse ambiente rico em possibilidades pode ser um desafio. Localizar obras específicas, verificar sua disponibilidade e acompanhar os empréstimos exige tempo e organização, o que pode dificultar tanto o trabalho dos responsáveis quanto a experiência dos alunos.

Pensando em otimizar esses processos, surge a proposta de um sistema digital de gestão bibliotecária. Essa ferramenta proporciona praticidade, permitindo que os alunos consultem o catálogo de qualquer lugar, façam reservas de forma rápida e acompanhem os prazos de devolução. Já para os administradores, o sistema centraliza o controle do acervo, automatiza registros de empréstimos e devoluções, e ainda fornece dados valiosos para melhorar a gestão da biblioteca.

Neste projeto, apresentamos uma solução moderna e acessível, desenvolvida para simplificar o cotidiano da comunidade escolar. O sistema busca integrar tecnologia e educação, promovendo uma experiência mais organizada, eficiente e convidativa para todos que utilizam a biblioteca.

O capítulo 2 trata de tal assunto, já o capítulo 3 descreve como ... o capítulo 4 por sua vez é sobre ... Escrevi de forma resumida e direta, faça de forma que o texto seja bom para ler

**2 REQUISITOS DO SISTEMA**

Os requisitos funcionais definem as funcionalidades e comportamentos que um sistema deve apresentar para atender às necessidades dos usuários. Eles descrevem o que o sistema deve fazer, como operações específicas, processamento de dados e interações com os usuários.

Esses requisitos são fundamentais para o desenvolvimento, pois orientam a equipe técnica e garantem que o sistema cumpra seus objetivos. Por exemplo, em um sistema de biblioteca, eles podem incluir funcionalidades como cadastro de livros, consulta de disponibilidade e registro de empréstimos.

Além de serem documentados de forma clara, os requisitos funcionais devem ser mensuráveis, permitindo validação e testes para assegurar que o sistema funcione conforme planejado.

“Todo capítulo precisa ter no mínimo três linhas”.

“Todo capítulo inicia em uma página”

2.1 Requisitos funcionais

“Toda tabela deve ter legenda”

|  |  |
| --- | --- |
| 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 |

**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).
* Campos a serem editados (colocar aqui).

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 – FR004 – Cadastrar Aluno**

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

Prioridade: Alta

Entrada:

* RA do aluno
* 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 alunos.

**2.2.5 – FR005 – 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).
* Campos a serem editados (colocar aqui).

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 alunos cadastrados.

**2.2.6 – RF006 – Excluir Aluno**

Descrição: O sistema deve permitir que o usuário exclua um aluno.

Prioridade: Média

Entrada:

* Identificação do aluno a ser excluído (ID).

Processamento:

* O sistema deve verificar se há empréstimos ativos ou pendentes associados ao livro.

**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:

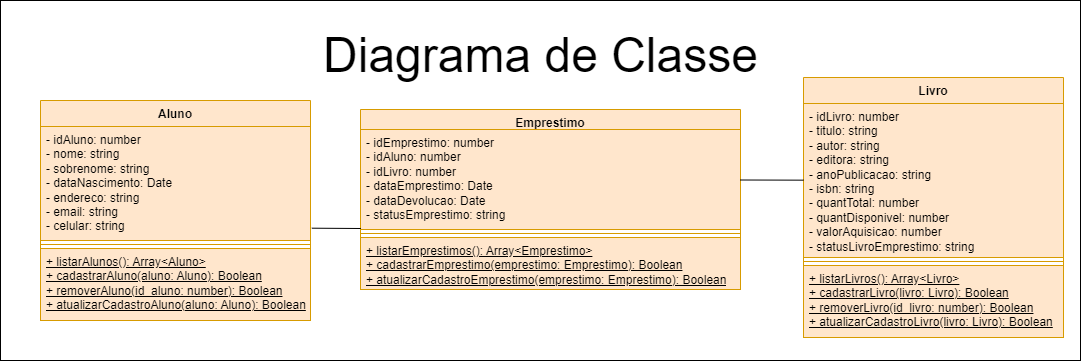
* O empréstimo deve ser registrado com status "Ativo".
* A quantidade disponível do livro deve ser atualizada no sistema.

**3 DIAGRAMAS DO SISTEMA**

Os diagramas são essenciais para programadores porque ajudam a visualizar a estrutura e o funcionamento do sistema antes de ser implementado. Eles mostram a relação entre os componentes, facilitando a compreensão do fluxo de informações. Além disso, servem como uma documentação clara, tornando mais fácil para outros desenvolvedores entenderem e darem manutenção no sistema.

3.1 Diagrama de Classe

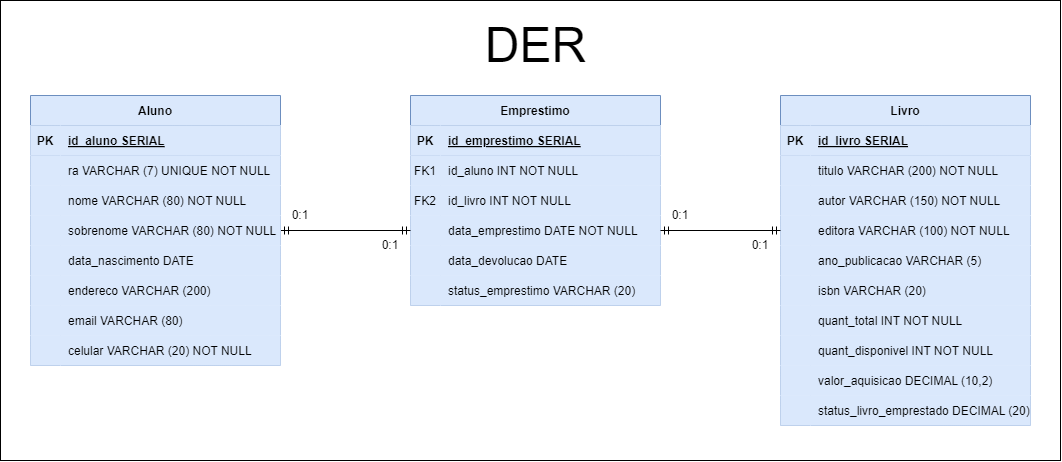
O diagrama de classe é essencial para o programador backend, pois define a estrutura do sistema, detalhando as classes, seus atributos e métodos, além das relações entre elas. Ele ajuda na organização do código e no planejamento da lógica de negócios, facilitando a implementação e a manutenção do software



3.2 Banco de Dados

O SGBD utilizado neste projeto é o MySQL, que é muito popular por sua confiabilidade e eficiência. Ele organiza os dados em tabelas, facilitando a consulta e manipulação por meio de SQL. Uma das principais características do MySQL é seu suporte a transações ACID, o que garante segurança e integridade dos dados. Além disso, ele permite o gerenciamento de grandes volumes de informações. Para a interface, podemos usar ferramentas gráficas, como o MySQL Workbench, ou a linha de comando, oferecendo flexibilidade para os desenvolvedores gerenciarem o banco de dados da maneira que preferirem

O diagrama DER (Diagrama Entidade-Relacionamento) é uma ferramenta visual importante que mostra como as diferentes entidades do banco de dados se relacionam. Ele facilita a compreensão da estrutura do sistema, ajudando os desenvolvedores a planejar e modelar o banco de dados de maneira eficiente. Além disso, funciona como um guia durante o desenvolvimento, ajudando a evitar erros e garantindo que tudo esteja bem organizado para a manutenção futura.



**4 ROTAS DA APLICAÇÃO – BACK-END**

Explica o que é rota, lembrando que nada nesse documento deve ser tratado como pergunta e resposta, cabe a contextualização ou o termo dissertação.

|  |  |  |
| --- | --- | --- |
| **Função** | **Método** | **Rota** |
| Cadastro de Aluno | POST | /novo/aluno |
| Cadastrar Livro | POST | /novo/livro |
| Listar Livro | GET | /listar/livro |
| Listar Aluno | GET | /listar/aluno |
| Listar Empréstimo | GET | /listar/emprestimo |

**5 INTERFACE E EXPERIÊNCIA DO USUÁRIO**

Uma interface é o meio de interação entre o usuário e um sistema ou dispositivo. Ela é composta por todos os elementos visuais e funcionais que permitem que o usuário execute ações e receba informações do sistema, como botões, menus, ícones, formulários e outros componentes. A interface pode ser gráfica, como em websites e aplicativos, ou de texto, como em sistemas de linha de comando.

O objetivo principal da interface é facilitar a interação, tornando-a intuitiva, eficiente e agradável. Uma boa interface deve ser fácil de entender e utilizar, garantindo que o usuário consiga atingir seus objetivos de maneira rápida e sem frustrações. A experiência do usuário depende fortemente da qualidade da interface, pois ela influencia diretamente a forma como o usuário se sente ao interagir com o sistema.







5.2 Paleta de Cores

A paleta de cores é um conjunto de cores escolhidas para um design, com o objetivo de criar harmonia visual e transmitir uma mensagem específica. Ela é composta por cores primárias, secundárias e neutras, que se combinam de forma a proporcionar clareza e estética. O uso adequado da paleta de cores é fundamental para garantir uma boa experiência ao usuário, influenciando a percepção e a legibilidade do projeto. Além disso, deve ser pensada para garantir acessibilidade, levando em consideração o contraste e a leitura para todos os usuários.

5.3 Mockup

1. Interface Inicial



Página - Inicio

1. Página da tabela com os livros cadastrados.



Página - Tabela de Livros

* 1. Página para cadastrar um livro



Página - Cadastrar Livro

2.2 Página para atualizar um livro.



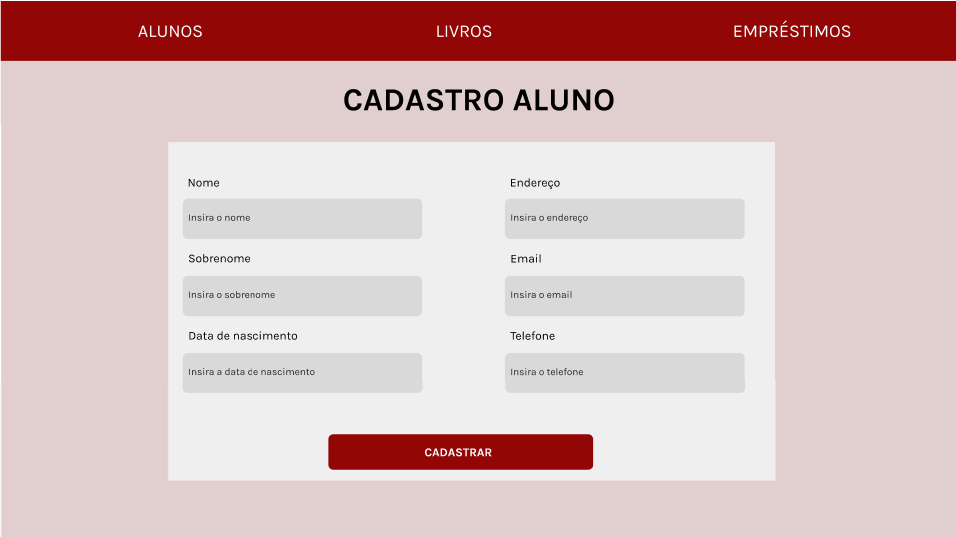
Página - Atualizar Livro

1. Página com a tabela dos alunos cadastrados



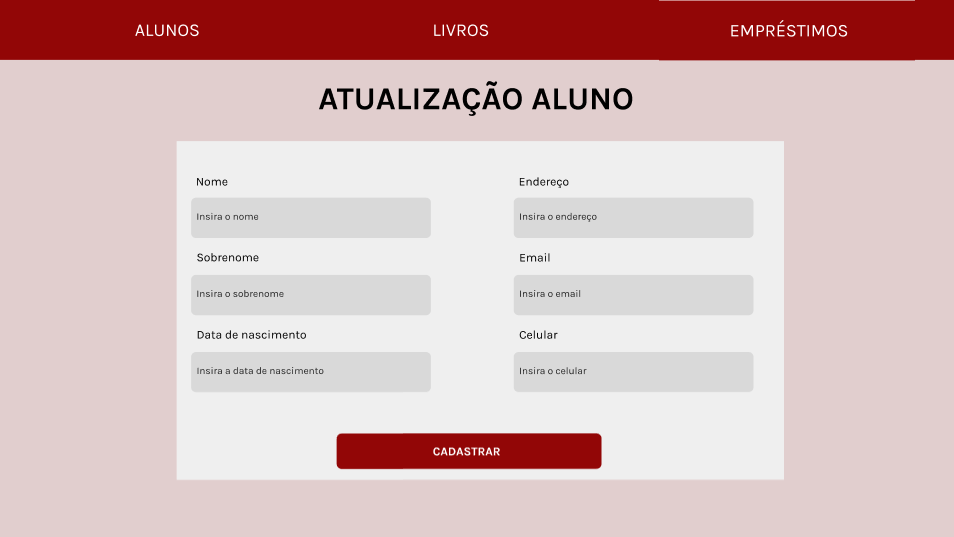
Página - Tabela de Alunos

3.1 Página de cadastrar um aluno.



Página - Cadastrar Aluno

3.2 Página de atualizar aluno.



Página - Atualizar Aluno

1. Página com a tabela de empréstimo de alunos.



Página - Tabela com Empréstimos

4.1 Página para cadastrar empréstimos.



Página - Cadastrar Emprestimo

4.2 Página de atualizar um empréstimo



Página - Atualizar Empréstimo

BIBLIOGRAFIA

Link das documentações utilizadas e livros consultados