**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Letícia Calixto Diniz**

**BiblioTechSistem**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

Antes do suporte da tecnologia, as bibliotecas organizavam seus arquivos, organizava o controle de empréstimos e devoluções manualmente, que causava um enorme atraso, além da enorme preservação dos livros e os cuidados com a documentação exigia muita cautela, já que qualquer dano ou perda poderia causar um enorme prejuízo. Esse processo também era muito distinto para os leitores, por exemplo, que para procurar um livro específico era preciso de muita paciência e assistência dos bibliotecários.

Nesse contexto, a BiblioTechSistem surge com o objetivo de modernizar as bibliotecas, avançando e facilitando sua utilização, transformando a experiência dos usuários por meio de uma plataforma moderna e que simplifica o gerenciamento dos acervos, histórico de leitura dos usuários e que possui um sistema de empréstimo e devolução automatizados e uma simples facilitação para que os leitores tenham acesso a livros não apenas como físicos mais também digitais com uma ampla variedade de gêneros literários.

**2 REQUISITOS DO SISTEMA**

A BilioTechSistem oferece requisitos essenciais que atende as necessidades dos usuários com eficiência e praticidade como por exemplo a adição e remoção de usuários, catálogos de livros e sistema de empréstimo e devolução, e entre outros.

**2.1 Requisitos funcionais**

Os requisitos funcionais descrevem o que o sistema deve fazer, incluindo suas funcionalidades e comportamentos esperados, no caso da BiblioTechSistem o sistema simplifica a permissão de reservas com notificações ou manter um histórico de reservas.

|  |  |
| --- | --- |
| 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
* ISNB
* Quantidade total.
* Quantidade disponível.
* 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 (título, autor, editora, ano de publicação, ISBN, quantidade total, quantidade disponível, valor da aquisição, status do livro emprestado).

Processamento:

* O sistema deve validar os campos obrigatórios.
* O sistema deve verificar se o ISBN ou ID editado não já existe no acervo (evitando duplicidade).

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\_livro);

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 na base de dados da biblioteca.

### **Prioridade**: Alta

### **Entrada**:

* Nome do aluno (obrigatório).
* Sobrenome (obrigatório).
* Data de nascimento. (obrigatório).
* Endereço.
* E-mail.
* Celular.

**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 matrícula.

**Pós-condição**:

* O aluno cadastrado deve ser exibido na lista de alunos disponíveis no sistema.

### **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\_aluno ou ra).

### Campos a serem editados (nome, sobrenome, data de nascimento, endereço, e-mail, celular).

**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 e nos registros de empréstimos associados ao aluno.

### **2.2.6** **RF006 – Excluir Aluno**

**Descrição**: O sistema deve permitir que o usuário exclua um aluno da base de dados da biblioteca.

**Prioridade**: Média

**Entrada**:

* **Identificação do aluno a ser excluído** (id\_aluno)

**Processamento**:

* + O sistema deve verificar se há empréstimos ativos ou pendentes associados ao aluno.
  + O sistema deve registrar o motivo da exclusão (opcional, mas recomendado para controle).

**Saída**:

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

**Pós-condição**:

* + O aluno deve ser removido da lista de alunos 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 são fundamentais no planejamento e desenvolvimento da aplicação, com o objetivo de mapear como o sistema se conecta e funciona como um todo, além de oferecerem uma visão clara dos requisitos e da estrutura do projeto de forma clara e eficiente.

**3.1 Diagrama de Classe**

Os diagramas são essenciais pois fornecem um ponto de vista simples de ser compreendido. A baixo a representação do diagrama de classe da biblioteca e suas principais entidades que são: Aluno, Empréstimo e Livro.

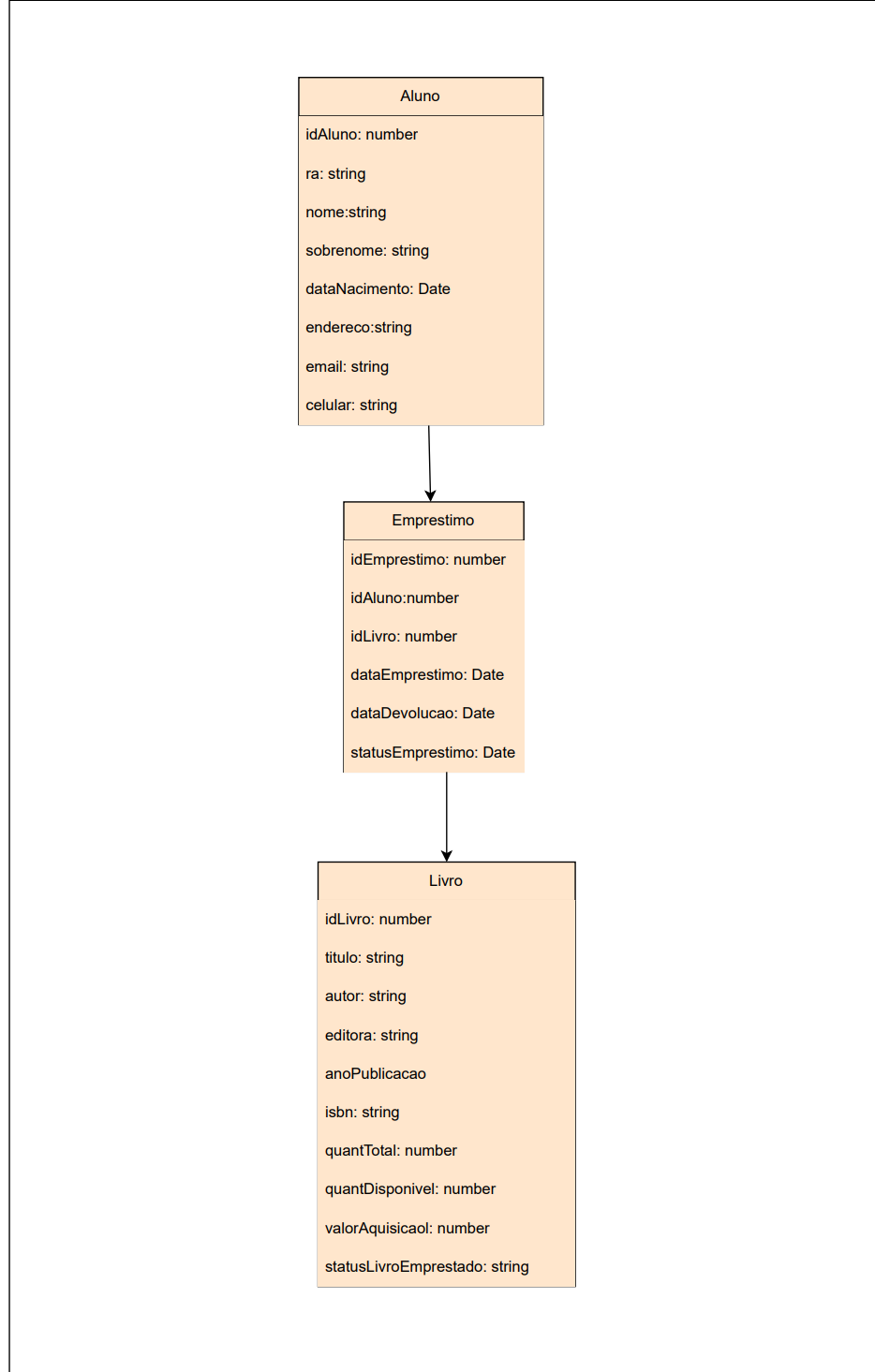


Figura 1 Imagem do Diagrama de classe

**3.2 Banco de Dados**

O DER é fundamental para as aplicações, pois assim como o diagrama acima também facilita e simplifica a visualização das entidades. Abaixo a representação do DER.

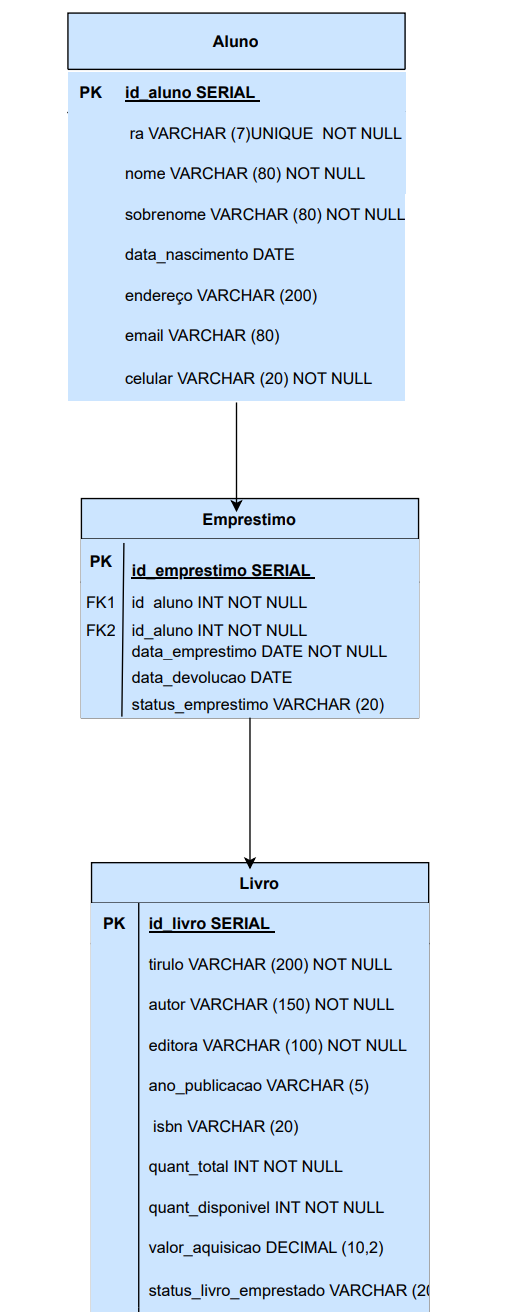
****

Figura 2 Imagem do DER da biblioteca

**4 ROTAS DA APLICAÇÃO – BACK-END**

As rotas são de extrema importancia para o desenvolvimento web, pois define o trajeto que da aplicação, elas conectam o front-end e o back-end e permitem que o servidor gerencie corretamente as requisições, isso inclui acessar informações, criar novos registros, deletar ou atualizar os dados no banco de dados.

|  |  |  |
| --- | --- | --- |
| **Função** | **Método** | **Rota** |
| Listar Aluno | GET | /alunos |
| Cadastrar Alunos | POST | /alunos |
| Deletar Alunos | DEL | /alunos |
| Atualizar alunos | PUT | /alunos |
| Listar Empréstimos | GET | /empréstimos |
| Cadastrar Empréstimos | POST | /empréstimos |
| Deletar Empréstimos | DEL | /empréstimos |
| Atualizar Empréstimos | PUT | /empréstimos |
| Listar Livros | GET | /livros |
| Cadastrar Livros | POST | /livros |
| Deletar Livros | DEL | /livros |
| Atualizar Livros | PUT | /livros |

**5 INTERFACE E EXPERIÊNCIA DO USUÁRIO**

A interface possui o objetivo de simplificar a interação com os usuários e a máquina, facilitando o uso e garantia de um sistema funcional e claro, permitindo que até usuários sem conhecimento de programação utilizem os aplicativos e plataforma facilmente.

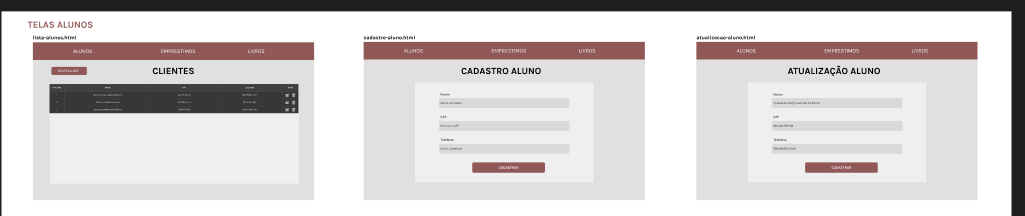
**5.2 Paleta de Cores**

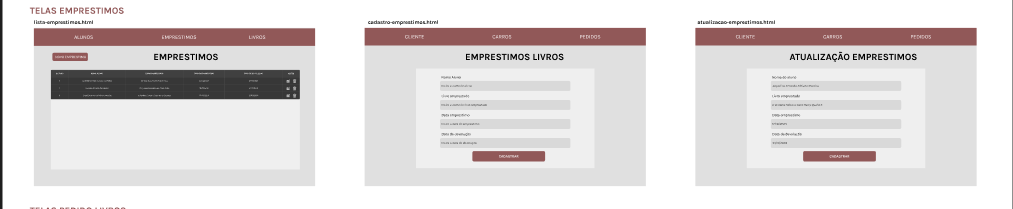
Decidi utilizar uma paleta de tons terrosos e tons de rosa, pois trazem uma imagem de aconchego e simplicidade, assim como o sentimento que a maioria das bibliotecas transmitem.

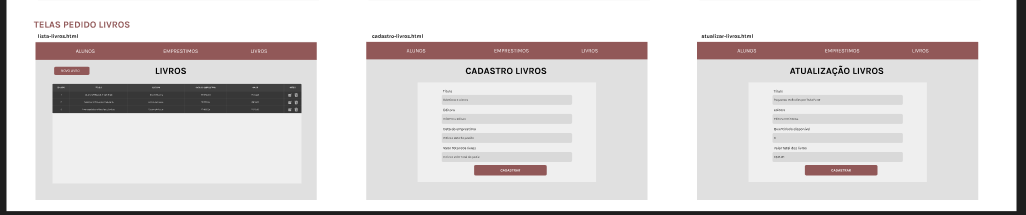
**5.3 Mockup**

A seguir, imagens do meu mockup.









BIBLIOGRAFIA

<https://miro.com/pt/diagrama/o-que-e-diagrama-entidade-relacionamento/>

<https://softdesign.com.br/blog/interface-e-usabilidade-para-desenvolvedores/#Qual-a-finalidade-do-uso-de-interfaces>

<https://softdesign.com.br/blog/requisitos-de-software-funcionais-e-nao-funcionais/#h-5-verificacao-e-aprovacao>