**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Claudio Gabriel Gonçalves batista**

**BIBLIOTECA**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

O gerenciamento tradicional de bibliotecas enfrentou dificuldades como o controle manual de empréstimos, organização complicada de acervos, dificuldade para encontrar livros e perda de registros.

E com a gestão digital, essas tarefas são automatizadas e mais simples. O sistema organiza o catálogo automaticamente, controla empréstimos em tempo real, envia notificações aos leitores e permite acesso remoto ao acervo. Isso torna a biblioteca mais ágil e melhor

**2 REQUISITOS DO SISTEMA**

Os requisitos do sistema são as condições que definem o que um software deve fazer e como ele deve operar. Eles incluem tantas funcionalidades quanto qualidades, como desempenho e segurança. Esses requisitos orientam o desenvolvimento para atender às expectativas

**2.1 REQUISITOS FUNCIONAIS**

Os requisitos funcionais são as funções que um sistema precisa realizar para cumprir seus objetivos. Eles definem o que o software deve fazer, como permitir login, cadastro ou realizar pagamentos. Essenciais no desenvolvimento, eles garantem que o sistema atenda às expectativas dos usuários

|  |  |
| --- | --- |
| 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 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 (obrigatorio)
* Ano de Publicação
* Isbn
* Quantidade total
* Quantidade disponível
* Valor de 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 (nome, autor, editor).

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:

* 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:

* nome (obrigatório).
* sobrenome (obrigatório).
* Data de nascimento
* Endereço
* Email
* 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 cadastro.

Pós-condição:

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

**2.2.5 RF005 – Editar Aluno**

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

Prioridade: Alta

Entrada:

* Identificação do aluno a ser editado ( ID ).
* Campos a serem editados (nome, sobrenome, cpf, endereço, email, 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 acervo e nos registros de alunos.

**2.2.6 RF006 – Excluir Aluno**

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

Prioridade: Média

Entrada:

* Identificação do aluno a ser excluído (ID).

Processamento:

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

Saída:

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

**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 no desenvolvimento de software porque facilitam a compreensão de ideias complexas, melhoram a comunicação entre a equipe e servem como documentação útil para a manutenção do sistema. Eles ajudam a identificar problemas antes da aprovação, permitem visualizar o impacto das mudanças e organizar o planejamento ao dividir tarefas grandes em partes menores

**3.1 Diagrama de Classe**

O diagrama de classe é uma ferramenta fundamental para programadores backend, pois fornece uma representação visual das classes, atributos e métodos de um sistema. Sua principal finalidade é ajudar na organização e no planejamento da estrutura do código, permitindo que o desenvolvedor visualize como as diferentes partes do sistema se relacionam

**3.2 Banco de Dados**

O PostgreSQL é um banco de dados relacional de código aberto, conhecido por sua robustez e suporte a dados complexos. O pgAdmin é uma interface gráfica popular que facilita a administração do PostgreSQL, permitindo criar e gerenciar bancos de dados de forma intuitiva.

O DER representa visualmente as entidades e relacionamentos de um banco de dados. Ele é essencial para garantir clareza, facilitar o planejamento e orientar a implementação, tornando-se fundamental para um design eficiente do sistema.

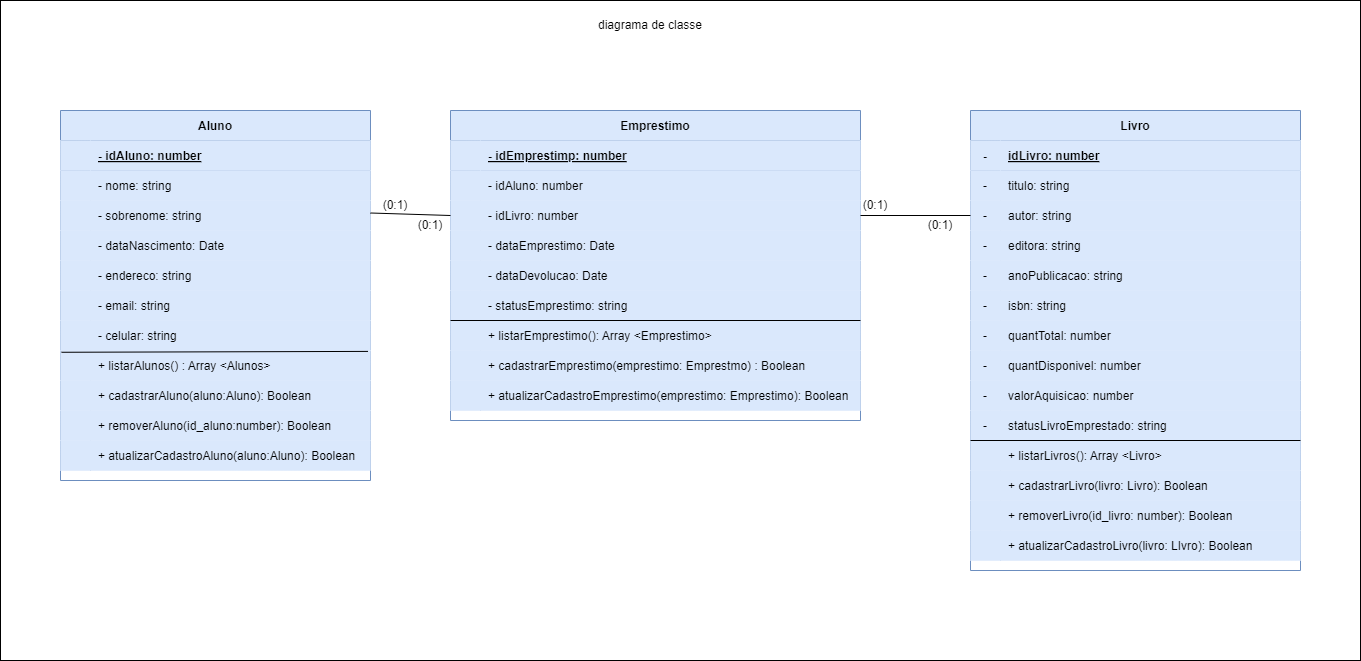


Figura diagrma de classe da biblioteca

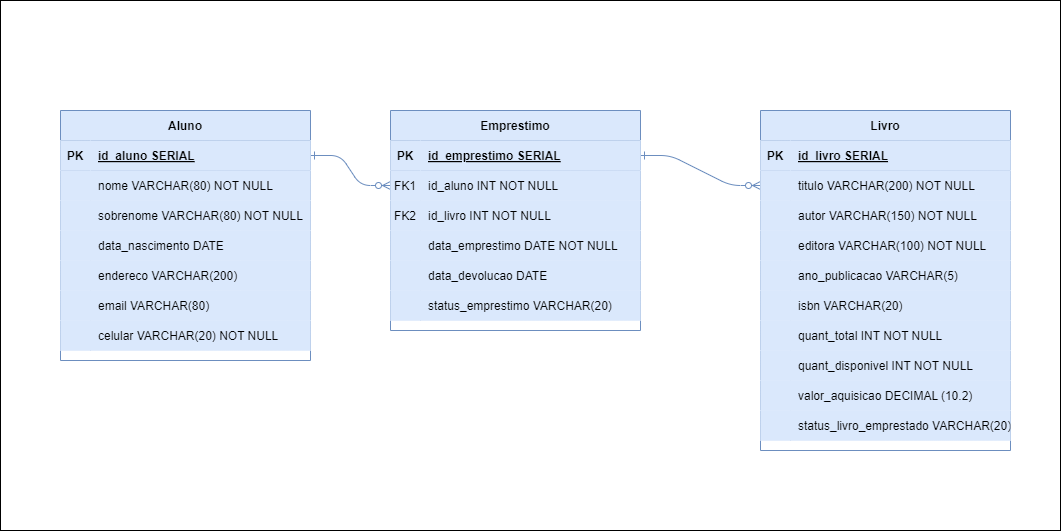


Figura DER biblioteca

**4 ROTAS DA APLICAÇÃO – BACK-END**

No desenvolvimento de software, as rotas de aplicação no back-end são caminhos que definem como um servidor responde a solicitações feitas por clientes, como navegadores ou aplicativos. Cada rota combina um método HTTP (GET, POST, PUT, DELETE) e um endpoint, que especifica o recurso ou ação desejada, como /usuarios ou /produtos/{id}. Elas são responsáveis por conectar a interface do cliente à lógica de negócio, processando dados enviados pela URL, query strings ou corpo da requisição. Por trás de cada rota, há uma função ou controlador que executa a lógica necessária, como acessar bancos de dados, validar informações ou gerar respostas. Em frameworks modernos, as rotas são organizadas de forma modular, permitindo clareza e eficiência no desenvolvimento. Essenciais para aplicações web e APIs, elas estruturam a comunicação entre cliente e servidor, garantindo que dados fluam de forma segura e consistente.

.

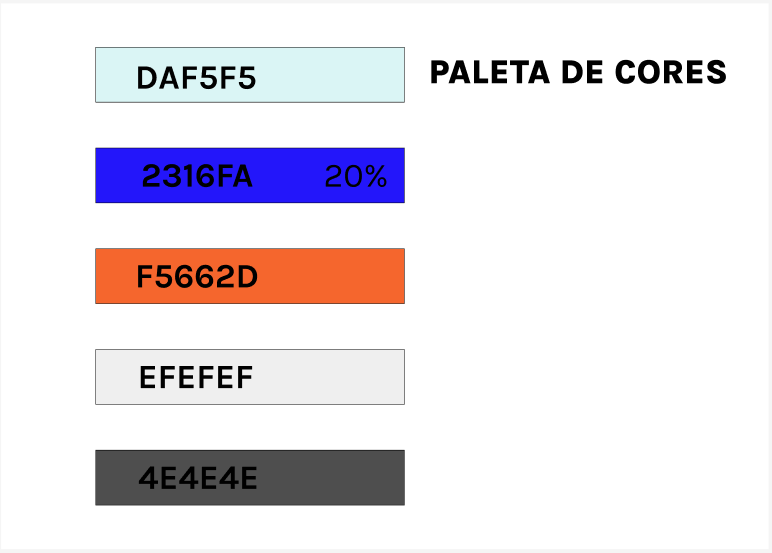
|  |  |  |
| --- | --- | --- |
| Listar aluno | GET | lista/alunos |
| Listar livro | GET | lista/livros |
| Listar emprestimo | GET | lista/emprestimo |
| Novo aluno | POST | novo/alunos |
| Novo livro | POST | novo/livros |

**5 INTERFACE E EXPERIÊNCIA DO USUÁRIO**

Uma **interface** é um meio de comunicação entre sistemas ou entre pessoas e dispositivos. O objetivo é facilitar essa interação, tornando-a simples e eficiente.

**5.2 Paleta de Cores**

Escreva sobre a paleta de cores e insira a imagem delas



**5.3 Mockup**

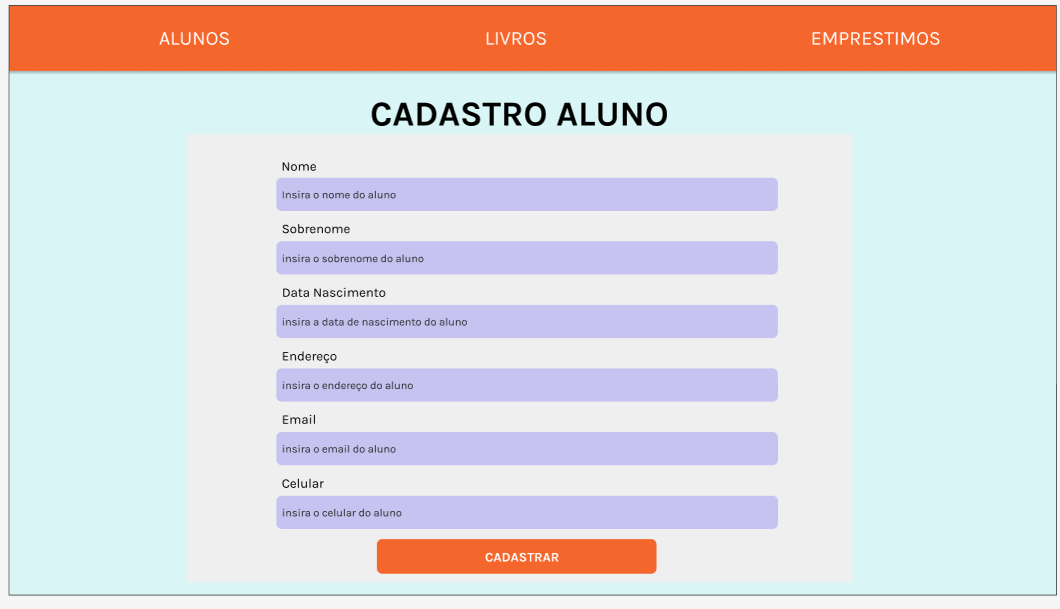
Texto antes da imagem, relatando a qual entidade aquela interface pertence

***1 – pagina principal***

***2 – lista de alunos***

******

2.1 – ***cadastro de aluno***

******

***2.2 – atualização de alunos***

******

***3 – lista de livros***

******

***3.1 – cadastro de livro***

******

***3.2 – atualização de livro***

******

***4 – lista de empréstimos***

******

***4.1 – cadastro de empréstimos***

******

***4.2 – atualização de empréstimos***

******

BIBLIOGRAFIA

Link das documentações utilizadas e livros consultados