**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Luís Eduardo Novaes Souza**

**Projeto Biblioteca**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

As bibliotecas escolares são uma janela para o saber e a criatividade, porém, a gestão do acervo pode ser um desafio. No sistema projeto biblioteca, a gestão digital se apresenta como uma alternativa revolucionária, simplificando procedimentos como a procura, o empréstimo e a devolução de livros. Através de uma interface intuitiva, os estudantes têm a capacidade de consultar a disponibilidade de títulos, fazer reservas de livros e administrar suas devoluções. Para os profissionais de biblioteca, a plataforma simplifica a gestão do acervo e fornece relatórios minuciosos sobre a utilização, possibilitando uma adaptação contínua às demandas da comunidade escolar. Ao conectar a biblioteca ao universo digital, essa abordagem não apenas aprimora a rotina diária, mas também torna a experiência de leitura mais interativa, personalizada e acessível, promovendo o aprendizado e o gosto pela leitura de maneira eficiente.

O capítulo 2 trata dos requisitos do sistema, já o capítulo 3 descreve como ... o capítulo 4 por sua vez é sobre ...

**2 REQUISITOS DO SISTEMA**

O sistema Projeto Biblioteca tem é necessário a utilização de requisitos do sistema no projeto para garantir que ele atenda às expectativas do usuário, funcione corretamente e cumpra suas funções. Eles orientam o desenvolvimento, evitam falhas e melhoram a eficiência do software, sendo obrigatórios ou não.

**2.1 Requisitos funcionais**

Dentro de requisitos do sistema, há os requisitos funcionais, a utilização deles é fundamental para garantir que o sistema execute suas funções principais corretamente. Pois dizem respeito as especificações, ações e comportamentos que o software deve realizar obrigatoriamente. Abaixo há a tabela de requisitos funcionais.

|  |  |
| --- | --- |
| Requisitos Funcionais | |
| 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).
* Quantidade total (obrigatório)
* Quantidade disponível (obrigatório)
* Ano de publicação
* ISBN
* Valor aquisição
* Status do Livro

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).
* Campos a serem editados (Título do livro; Autor(es); Editora; Quantidade Total; Ano de publicação; Valor de aquisição; ISBN; Status do Livro).

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 o cadastro de novos alunos no sistema de gerenciamento da biblioteca.

Prioridade: Alta

Entrada:

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

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 gerenciamento disponível.

**2.2.5 RF005 – Editar Aluno**

Descrição: O sistema deve permitir que o Aluno tenha suas informações editadas.

Prioridade: Alta

Entrada:

* Identificação do aluno a ser editado (ID ou RA).
* Campos a serem editados (Nome; RA; Sobrenome; Celular; Data de nascimento; Endereço; E-mail).

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 gerenciamento e nos registros de empréstimos associados ao aluno.

**2.2.6 RF006 – Excluir Aluno**

Descrição: O sistema deve permitir que o Aluno seja excluído do gerenciamento.

Prioridade: Média

Entrada:

* Identificação do aluno a ser excluído (ID ou RA).

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 usuário.

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 gerenciamento e não será mais possível ter empréstimos associados a ele.

**2.2. 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 para todos os programadores, eles ajudam a visualizar a estrutura do sistema, facilitam o entendimento de como diferentes componentes interagem, identificam possíveis problemas de design, otimizam a arquitetura e evitam erros antes da implementação além promoverem uma comunicação clara entre a equipe de desenvolvimento e outras partes interessadas, garantindo que todos compreendam a funcionalidade do sistema.

**3.1 Diagrama de Classe**

No desenvolvimento de software orientado a objetos, o diagrama de classes é essencial, pois ele estabelece a estrutura do sistema, apresentando as classes, seus atributos, métodos e as conexões entre elas. Auxiliando os desenvolvedores no planejamento da estruturação do código, assegurando a consistência e a reutilização de elementos. Ademais, simplifica a interação entre os programadores, possibilitando que todos entendam a lógica do sistema, além de atuar como registro do projeto, sendo benéfico para futuras manutenções e ampliações do software. Na figura 1 há representação do diagrama de classe.

**3.2 Banco de Dados**

O pgAdmin é uma interface gráfica que torna a administração do PostgreSQL mais simples, permitindo criar tabelas, executar consultas e gerenciar permissões de forma intuitiva, sem precisar utilizar o terminal.

O Diagrama de Entidade-Relacionamento (DER) representa a estrutura do banco de dados, destacando entidades, atributos e relações. Ele assegura que o sistema cumpra as exigências de dados e simplifica a interação entre programadores e partes interessadas. O DER também é a fundação para a elaboração do banco, auxiliando na previsão de problemas de design e no aprimoramento da eficiência. Segue o DER na figura 2.

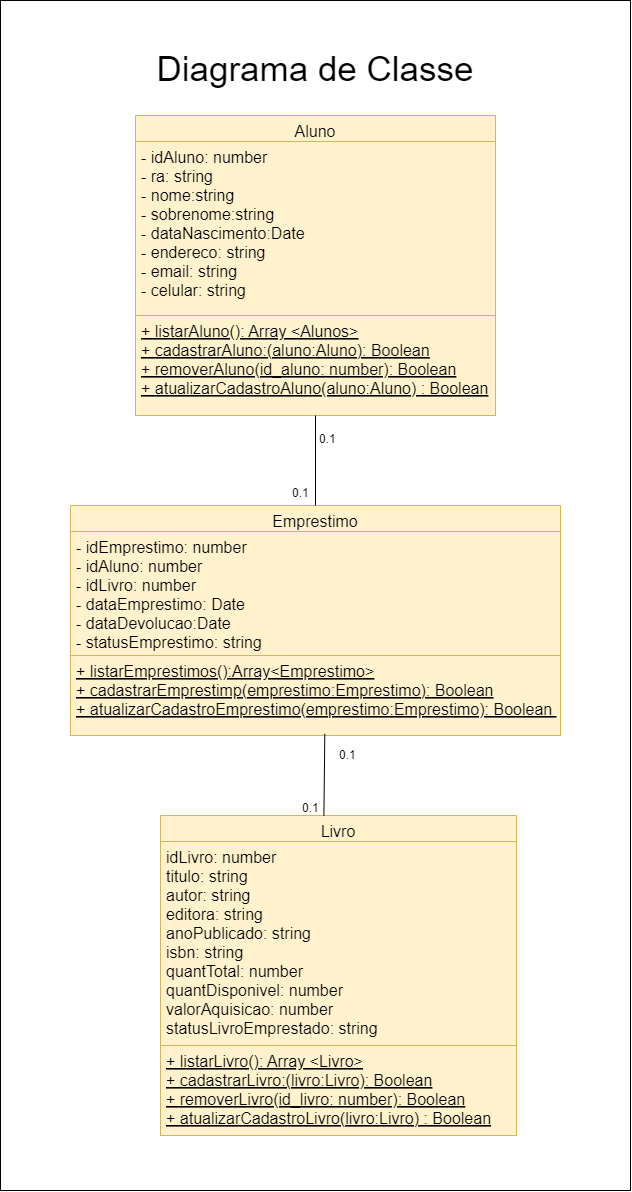


Figura 1 - DIAGRAMA DE CLASSE

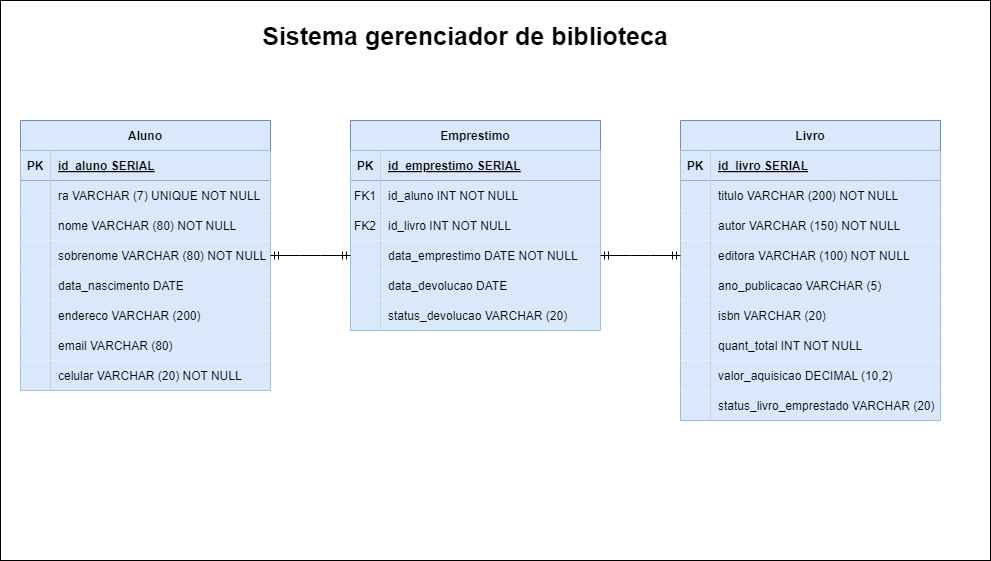


Figura 2 - DER

**4 ROTAS DA APLICAÇÃO – BACK-END**

No desenvolvimento de sistemas, as rotas desempenham um papel crucial ao definir os caminhos que conectam o front-end às funcionalidades do back-end, permitindo que o servidor interprete e responda às requisições feitas pelos usuários. Elas possibilitam operações como consulta, cadastro ou modificação de informações no banco de dados. No projeto biblioteca, as rotas estruturam funcionalidades essenciais, como listagem, cadastro, atualizar e deletar alunos, livros e empréstimos. A Tabela 2 detalha as principais rotas implementadas no sistema, com suas funcionalidades, assegurando a organização e eficiência da aplicação. Essas rotas garantem o fluxo de informações entre cliente e servidor, sendo fundamentais para a escalabilidade e o correto funcionamento do sistema.

|  |  |  |
| --- | --- | --- |
| **Função** | **Método** | **Rota** |
| listagemAlunos | GET | /lista/alunos |
| cadastroAluno | POST | /novo/aluno |
| listagemLivro | GET | /lista/livros |
| cadastroLivro | POST | /novo/livro |
| listagemEmprestimos | GET | /lista/emprestimos |

Tabela 2- ROTAS DE APLICAÇÃO

**5 INTERFACE E EXPERIÊNCIA DO USUÁRIO**

A interface é o meio pelo qual ocorre a interação entre sistemas, dispositivos ou usuários, e pode ser classificada em interface de usuário (UI) e interface de programação (API). A UI envolve o design visual e interativo de um software, permitindo ao usuário executar ações diretamente no sistema. Já a API viabiliza a comunicação entre diferentes módulos ou aplicações, promovendo troca de dados e integração. Sua principal finalidade é simplificar processos e garantir eficiência. Interfaces de usuário bem projetadas tornam sistemas acessíveis e intuitivos, otimizando a experiência e a produtividade do usuário.

**5.1 Paleta de Cores**

A paleta de cores do projeto Biblioteca foi cuidadosamente planejada para transmitir uma atmosfera acolhedora e otimista. A cor principal, um vibrante amarelo-laranja, simboliza calor e positividade, criando uma conexão emocional envolvente com o usuário. Para equilibrar e destacar essa tonalidade, foram selecionadas cores neutras, como branco e preto, aplicadas em elementos fundamentais da interface, como fundos, textos, botões e formulários. Essa combinação harmônica não apenas realça a energia do amarelo-laranja, mas também oferece uma experiência visual limpa, minimalista e agradável, promovendo conforto e funcionalidade, como mostrado na figura 3.



Figura 3- PALETA DE CORES

Hexadecimal e RGB das cores utilizadas:

Amarelo-laranja vibrante: #FBB02D / (251, 176, 45);

Branco: #FFFFFF / (255, 255, 255);

Preto: #000000 / (0, 0, 0);

**5.2 Mockup**

BIBLIOGRAFIA

Link das documentações utilizadas e livros consultados