**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Yuri Dinato da Silva**

**Biblioteca Automatizada**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

Visualize uma biblioteca onde a organização e o acesso a obras e usuários sejam tão intuitivos quanto o ato de folhear um livro. Com nosso sistema de gestão inovador, essa ideia se torna concreta, proporcionando uma experiência fluida para todos os participantes. O sistema oferece um registro personalizado para obras e usuários, com informações detalhadas que simplificam a busca e a administração. A atualização de informações é ágil e as opções de pesquisa são variadas, possibilitando uma visualização eficaz do acervo e dos empréstimos, com alertas automáticos que previnem atrasos. Tanto estudantes quanto bibliotecários se beneficiam, pois os estudantes localizam obras com facilidade e monitoram seus empréstimos, enquanto os bibliotecários têm suas atividades automatizadas, permitindo um maior foco na promoção da leitura.

Os principais ganhos para a biblioteca envolvem a eficiência dos procedimentos, o aprimoramento da estrutura e a contentamento dos frequentadores, tudo isso favorecendo um espaço mais acessível e propício à leitura. Aposte em um futuro mais inteligente para sua biblioteca. Com nossa solução, ela se tornará um local mais bem estruturado, produtivo e convidativo. Fale conosco e veja como podemos auxiliar na transformação de sua biblioteca em um ambiente ainda mais inspirador!

**2 REQUISITOS DO SISTEMA**

Requisitos funcionais constituem uma parte essencial na criação de software e sistemas. Eles detalham as funcionalidades e comportamentos específicos que o sistema deve apresentar para satisfazer as demandas dos usuários ou das partes interessadas. Esses requisitos definem o que o sistema deve realizar em termos de ações e operações, sendo cruciais para o desenvolvimento do projeto. No caso do sistema, cadastro de usuários e livros, empréstimos e devoluções são requisitos do sistema.

**2.1 Requisitos funcionais**

Requisitos funcionais são descrições que mostram as funcionalidades e atributos que um sistema ou software deve ter para satisfazer as demandas dos usuários ou partes interessadas. Eles estabelecem as atividades que o sistema deve executar, como reagir em diversas circunstâncias, o que deve ser processado e de que maneira interagir com os usuários ou outros sistemas. Esses requisitos detalham o que o sistema realiza, abrangendo ações como cadastrar usuários, criar relatórios e gerenciar transações. Ademais, definem como o sistema deve reagir às solicitações dos usuários e quais recursos devem estar disponíveis.  Esses requisitos são fundamentais para orientar o desenvolvimento e garantir que o sistema cumpra as expectativas funcionais dos stakeholders.

|  |  |
| --- | --- |
| 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 publicação
* Isbn
* Quantidade total (obrigatório).
* Quantidade disponível (obrigatório).
* 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 (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 RF004 – Cadastrar Aluno**

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

Prioridade: Alta.

Entrada:

* Nome do aluno (obrigatório).
* RA (obrigatório).
* Nome (obrigatório).
* Sobrenome (obrigatório).
* Data de nascimento.
* Endereço.
* Email
* 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 cadastrados.

**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 ou RA).
* 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 e nos registros de empréstimos associados ao aluno.

**2.2.6 RF006- Excluir Aluno**

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

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 excussão.

Pós-condição:

* O livro deve ser removido da lista de cadastro 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:

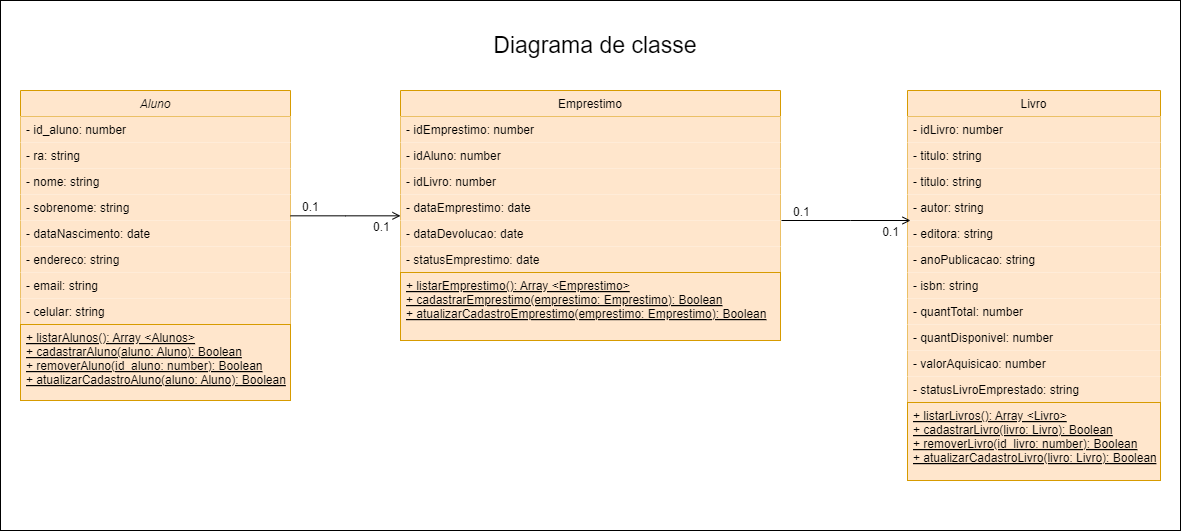
* O empréstimo deve ser registrado com status "Ativo".
* A quantidade disponível do livro deve ser atualizada no sistema.

**3 DIAGRAMAS DO SISTEMA**

Os diagramas são fundamentais para os programadores, uma vez que auxiliam na compreensão da estrutura do sistema, simplificam a comunicação entre os membros da equipe e estruturam o processo de trabalho. Em relação a um sistema de administração de bibliotecas, eles possibilitam o mapeamento das conexões entre entidades como "Livro", "Aluno" e "Empréstimo", além de aprimorar o entendimento de processos, como empréstimos e devoluções. Os diagramas também minimizam falhas no código e simplificam a manutenção e as atualizações futuras do sistema, assegurando uma implementação mais eficaz e estruturada.

**3.1 Diagrama de Classe**

O diagrama de classes organiza as partes do sistema, como "Livro", "Usuário" e "Empréstimo", mostrando suas relações. Auxiliando o programador na hora do back-end, na criação dos atributos e métodos.

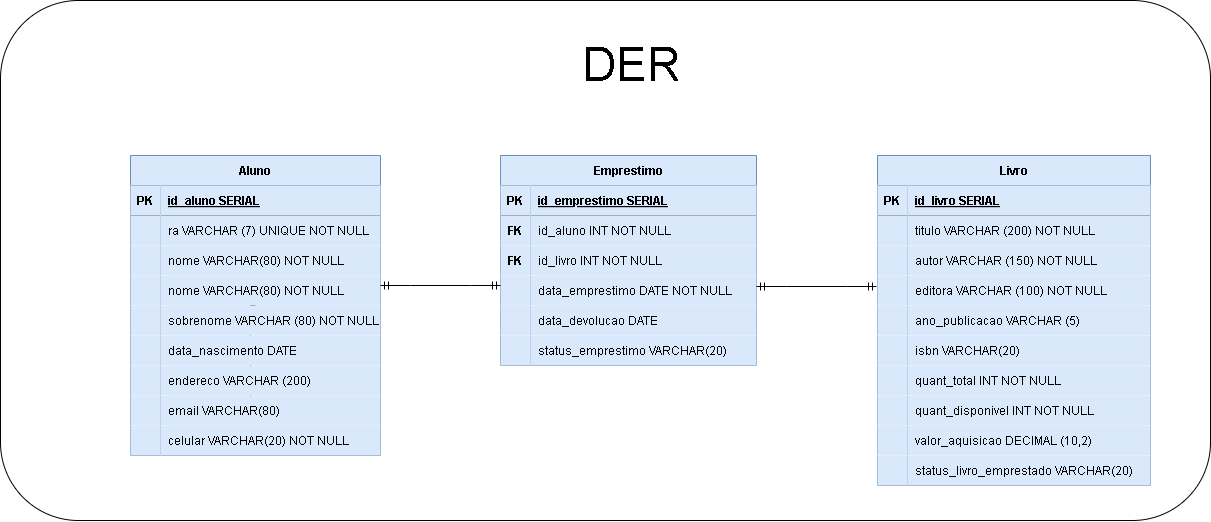


**3.2 Banco de Dados**

O pgAdmin é um aplicativo visual para administração de bases de dados PostgreSQL, proporcionando uma interface intuitiva para administração, realização de consultas e acompanhamento de performance.

O Diagrama Entidade-Relacionamento (DER) é crucial para planejar e estruturar a estrutura de um banco de dados, particularmente em sistemas de administração de bibliotecas, assegurando eficácia, integridade e clareza nas conexões entre os dados.

Usamos os scripts CREATE TABLE, para criar as tabelas: empréstimo, aluno e livro, dentro do CREATE colocamos os atributos de cada tabela, atributos são as características. Usamos o INSERT para inserir valores dentro de cada tabela, adicionar alunos, livros e empréstimos.



**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.

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

5 INTERFACE E EXPERIÊNCIA DO USUÁRIO

Escreva o que é uma interface e o objetivo dela

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

BIBLIOGRAFIA

Link das documentações utilizadas e livros consultados