**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**João Gabriel Sabino De Araújo**

**BOOKHUB**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

A biblioteca escolar é um local onde os estudantes podem adquirir novos conhecimentos e melhorar suas habilidades de leitura. Contudo, administrar este ambiente, particularmente a organização de livros, sua localização e monitoramento, torna-se uma tarefa extremamente monótona para o profissional de biblioteca e os usuários. Verificar manualmente a existência de livros e coleções é uma tarefa que grande dedicação.

Um sistema de gestão de biblioteca digital é extremamente importante, pois proporciona uma boa solução para localizar livros e manter a coleção. Os alunos irão acessar facilmente o catálogo que ajuda a organizar suas sessões de leitura. Os bibliotecários podem ter controle total sobre o gerenciamento de estoque, registrando transações e gerando relatórios para dar suporte à instituição. Sempre garanta que o design gire em torno de um sistema central intuitivo que aprimore a experiência individual para promover um relacionamento mais rico e ativo entre usuários e bibliotecas.

**2 REQUISITOS DO SISTEMA**

Os requisitos do sistema são essenciais para a criação de um projeto, pois garantem que o software atenda às expectativas dos usuários e funcione de maneira correta. Eles orientam todo o processo de desenvolvimento, assegurando que as funcionalidades desejadas sejam implementadas de forma eficaz. Além disso, os requisitos ajudam a prevenir erros e a melhorar a eficiência do software. Essa classificação pode incluir tanto os requisitos obrigatórios, que são imprescindíveis, quanto os opcionais, que podem agregar valor ao sistema.

**2.1 Requisitos funcionais**

Os requisitos de funcionais definem as funções e atividades específicas que o software deve executar. Eles estabelecem as respostas que o sistema deve oferecer a entradas determinadas, garantindo que o comportamento do software esteja alinhado com as expectativas dos usuários durante sua utilização. Esses requisitos são essenciais para assegurar que todas as interações dentro do sistema ocorram de maneira eficaz e conforme o planejado.

|  |  |
| --- | --- |
| Requisito Funcional | |
| RF001 | Cadastrar Livro |
| RF002 | Editar Livro |
| RF003 | Remover Livro |
| RF004 | Cadastro de aluno |
| RF005 | Edição de aluno |
| RF006 | Remover alunos |
| 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\_publicacao
* Isbn
* Quant\_total(obrigatório)
* Quant\_disponivel (obrigatório)
* Valor\_aquisição
* Status\_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 do livro (obrigatório).

• Autor(es) (obrigatório).

• Editor (obrigatório).

• Ano publicação

• isbn

• Quantidade total (obrigatório).

• Quantidade disponível (obrigatório).

• Valor aquisição

• Status livro emprestestado

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 gerenciamento da biblioteca.

Prioridade: Alta

Entrada:

* Nome (obrigatório)
* Sobrenome (obrigatório)
* Data 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 acervo disponível.

**2.2.5 RF005 – Editar Aluno**

Descrição: O sistema deve permitir que o usuário edite as informações de um livro previamente cadastrado.

Prioridade: Alta

Entrada:

Campos a serem editados (Título do livro(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 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.

**2.2.6 RF006 – Excluir Aluno**

Descrição: O sistema deve permitir que o usuário exclua um livro 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 Alunos.
* 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 lAluno deve ser removido da lista de acervo 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 importantes para os programadores porque eles ajudam a entender como o sistema funciona. E também tornam o planejamento mais fácil, permitindo que os programadores visualizem tudo antes de começar a codificar, o que ajuda a evitar alguns problemas. Além disso, servem como um guia para iniciantes, facilitando a manutenção e as atualizações de um sistema.

**3.1 Diagrama de Classe**

O diagrama de classes é muito útil para programadores backend porque ajuda a visualizar as entidades do sistema e como elas se relacionam. Ele torna mais fácil entender a estrutura dos dados e as interações entre as classes, o que facilita o desenvolvimento e a manutenção do código.

**3.2 Banco de Dados**

Neste projeto, utilizamos o PostgreSQL, um sistema de gerenciamento de banco de dados relacional conhecido por sua confiabilidade e recursos avançados. Ele suporta diversos tipos de dados, incluindo JSON, e oferece transações seguras e alta concorrência, tornando-o ideal para aplicações que precisam lidar com grandes volumes de dados.

Diagrama DER

O Diagrama Entidade-Relacionamento (DER) é importante para a aplicação porque mostra de forma clara as entidades e seus relacionamentos no banco de dados. Isso ajuda a entender como as informações estão organizadas, facilitando o design e a implementação.

**4 ROTAS DA APLICAÇÃO – BACK-ENDs**

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

R: A rota em um sistema web é muito importante e é responsável por mapear a solicitações "HTTP que são GET, POST, PUT e o DELETE" para funções específicas que usam o backend. Eles estabelecem o caminho entre o cliente e o servidor, determinando qual lógica de negócio será executada e além disso, eles organizam e tornam acessíveis todos os recursos do aplicativo.

|  |  |  |
| --- | --- | --- |
| FUNCAO | MÉTODO | ROTA |
| ListarAluno | GET | /listar/aluno |
| ListarLivro | GET | /listar/livro |
| ListarEmprestimo | GET | /listar/emprestimo |
| CadastrarAluno | POST | /cadastrar/aluno |
| CadastrarLivro | POST | /cadastrar/livro |

**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