**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Nathália Caroline Gumbio**

**BiblioSmart**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

A biblioteca escolar estava passando por desafios em sua administração, livros sendo devolvidos após o prazo, o bibliotecário sem controle do acervo, saber sobre a disponibilidade de algum título eram tarefas complicadas. Para isso a direção decidiu propor como solução a criação de um sistema gerenciável da biblioteca para ajudar na organização, gerenciamento e otimização dos recursos da biblioteca. Com isso entraram em contatos com empresas para verem as soluções propostas por cada uma, com isso a empresa Librarix apresentou a ideia da BiblioSmart, um sistema de gerenciamento da biblioteca que pode conter os atuais problemas.

Com esse sistema os processos de catalogação, empréstimo, devolução e renovação dos livros serão otimizados e possibilitarão um melhor funcionamento, além de garantir o gerenciamento de usuários e o controle do acervo. Com o sistema automatizado, o bibliotecário pode registrar rapidamente o empréstimo de um item, associando-o ao cadastro do usuário. O sistema controla os prazos de devolução, evita o empréstimo simultâneo do mesmo material e emite alertas para renovações ou devoluções em atraso. O usuário pode fazer a consulta do acervo de forma remota, através de interfaces online, ele pode buscar por títulos específicos, autores ou até mesmo temas, com a ajuda de filtros avançados de pesquisa.

Este arquivo apresenta informações sobre como esse sistema ajudará tanto o bibliotecário quanto os alunos, promovendo uma maior integração entre os recursos da biblioteca e as necessidades dos seus usuários.

**2 REQUISITOS DO SISTEMA**

Os requisitos são essenciais no processo de desenvolvimento, pois orientam a construção do software e garantem que atenda às necessidades dos usuários e aos objetivos do negócio. No sistema da BiblioSmart têm-se o cadastro

Em um sistema, por exemplo, de gerenciamento de biblioteca, a definição clara e precisa dos requisitos é fundamental para que o sistema atenda às expectativas de usuários, bibliotecários e gestores, como garantir que dados de empréstimos não sejam perdidos durante falhas, contribuindo para uma operação eficiente e satisfatória.

**2.1 Requisitos funcionais**

Na tabela 1 está identificado os requisitos funcionais do sistema BiblioSmart, como funcionalidades e operações, ou seja, como o sistema deve agir em resposta a certas entradas ou ações, para atender as expectativas dos usuários e da gestão da biblioteca.

|  |  |
| --- | --- |
| 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 (não obrigatório)
* Isbn (não obrigatório)
* Quantidade total (obrigatório)
* Quantidade disponível (obrigatório)
* Valor aquisição (não obrigatório)
* Status do livro emprestado (não obrigatório)

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 (autor(es), editora, ano de publicação, isbn, quantidade total, quantidade disponível, valor aquisição, status do livro emprestado).

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.

Prioridade: Alta

Entrada:

* Ra (obrigatório).
* Nome (obrigatório).
* Sobrenome (obrigatório).
* Data de nascimento (não obrigatório).
* Endereço (não obrigatório).
* Email (não obrigatório)
* Celular (obrigatório)

Processamento:

* O sistema verifica se os campos obrigatórios foram preenchidos.
* O sistema deve validar se já não existem usuários com o mesmo ra.

Saída:

* Mensagem de sucesso ao cadastrar o Aluno.
* Mensagem de erro caso algum problema tenha sido detectado.

Pós-Condição:

* O aluno é cadastrado e seu usuário criado no sistema.

**2.2.5 RF005 – Editar Aluno**

Descrição: O sistema permitir que o usuário possa editar o aluno.

Prioridade: Alta

Entrada:

* Identificar o usuário que será editado (ID ou RA)
* Campos a serem editados: (ra, nome, sobrenome, data de nascimento, endereço, email, celular).

Processamento:

* O sistema verifica os campos obrigatórios.
* O sistema valida as alterações.

Saída:

* Mensagem de sucesso ao editar o usuário.
* Mensagem de erro caso alguma informação esteja incorreta.

Pós-condição:

* As alterações são salvas e o usuário é atualizado.

**2.2.6 RF006 – Excluir Aluno**

Descrição: O sistema deve permitir a exclusão de usuários.

Prioridade: Alta

Entrada:

* Identificar o usuário a ser excluído (ID ou RA)

Processamento:

* O sistema verifica se há algum livro emprestado no nome do usuário, caso exista a exclusão não é permitida.
* O sistema pede a confirmação de que o aluno realmente quer excluir o usuário.

Saída:

* Mensagem de sucesso ao excluir o usuário
* Mensagem de erro caso exista algum empréstimo no nome do usuário.

Pós-Condição:

* O usuário é excluído do sistema.

**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**

**3.1 Diagrama de Classe**

O diagrama de classe é uma ferramenta fundamental que serve como um mapa visual que representa as classes do sistema BiblioSmart, os seus atributos, métodos e relacionamentos. Ele permite que os programadores planejem o desenvolvimento de forma mais eficaz, facilita a manutenção do código e também a comunicação entre os membros da equipe. Modelar o sistema antes da implementação ajuda a identificar problemas de design e arquitetura, ajudando no funcionamento do sistema e aprimorando melhor o funcionamento da BiblioSmart. A figura 1 representa o diagrama de classe do sistema da BiblioSmart.

**3.2 Banco de Dados**

O **PostgreSQL** é um sistema de gerenciamento de banco de dados relacional de código aberto, conhecido por sua robustez, flexibilidade e conformidade com padrões, sendo ele o utilizado no sistema da BiblioSmart como Sistema de Gerenciamento de Banco de Dados. É amplamente utilizado em aplicações que exigem um alto nível de desempenho e integridade dos dados, ele garante propriedades ACID assegurando que as transações sejam tratadas de forma confiável e utiliza da ferramenta pgAdmin como interface gráfica, que facilita o gerenciamento do banco de dados, assim o usuário pode criar tabelas, executar queries, monitorar o desempenho, entre outras funções.

O Diagrama Entidade-Relacionamento (DER) é uma representação visual que ilustra as entidades do sistema, seus atributos e os relacionamentos entre elas. Ele é fundamental para o entendimento da estrutura do banco de dados e serve como um guia para o desenvolvimento do modelo físico que ajuda a, por exemplo, planejar a estrutura do banco de dados antes da implementação, garantindo que todos os requisitos sejam atendidos. Isso está sendo apresentado na figura 2.

Os scripts de banco de dados são fundamentais para a criação, modificação e manutenção da estrutura do banco de dados, assim como o PostgreSQL, usados para facilitar e automatizar tarefas, por exemplo, entre outras funções.

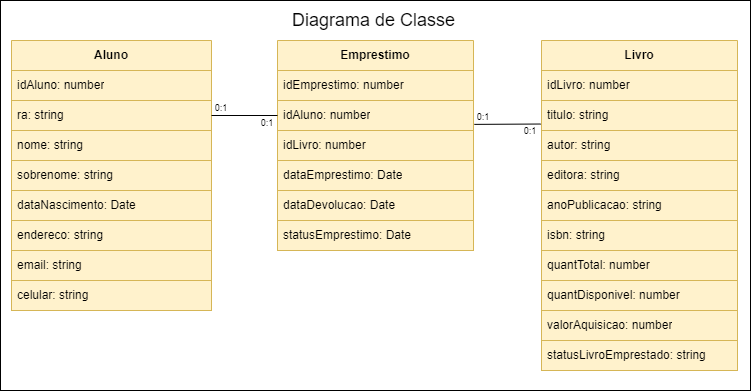
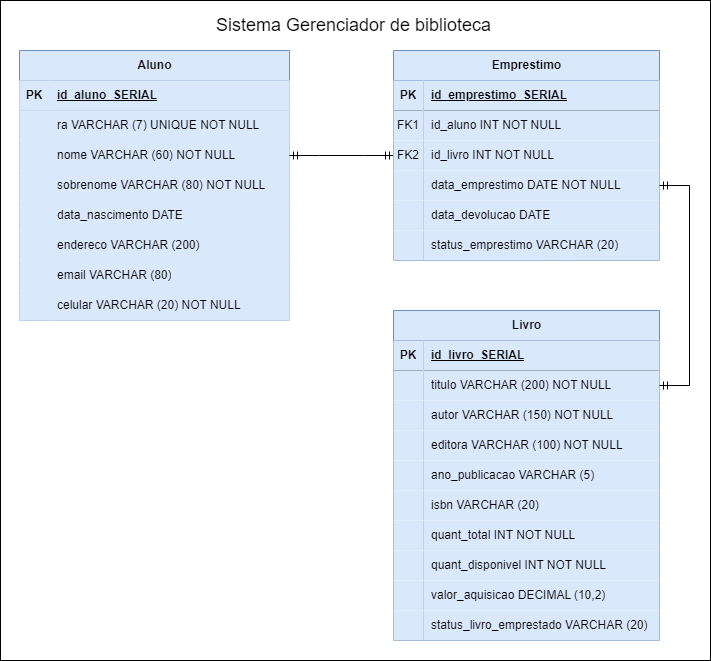


Figura 2-DER BiblioSmart

Figura 2-Diagrama de Classe BiblioSmart

**4 ROTAS DA APLICAÇÃO – BACK-END**

Rotas de aplicação, dentro do contexto de back-end, possuem a função de se referir a um caminho específico definido na API, o que determina as solicitações HTTP precisam ser tratadas. Para que funcione, cada rota precisa ser associada a um recurso e mapeia uma URL, a uma função que lide com a lógica para essa solicitação. A URL quando define o caminho, pode incluir parâmetros dinâmicos para, por exemplo, acessar um usuário específico. Por fim, as rotas também ajudam a organizar o código separando a lógica das outras partes da aplicação, tornando ela mais fácil de entender e manter o código limpo e funcionando.

|  |  |  |
| --- | --- | --- |
| GET | /lista/alunos | Lista os alunos cadastrados. |
| POST | /novo/aluno | Cadastra um novo aluno. |
| GET | /lista/livros | Lista os livros cadastrados. |
| POST | /novo/livro | Cadastro de um novo livro. |
| GET | /lista/emprestimo | Lista de empréstimos cadastrados no sistema. |

Tabela 2 - Rotas do código

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