**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Ana Júlia Souza Gomes**

**PROJETO GB- Biblioteca**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

**1 INTRODUÇÃO**

A importância de um sistema digital na organização das bibliotecas. A administração de um acervo é uma tarefa que exige tempo e atenção, tanto por partes dos bibliotecários quanto dos alunos. Desde a busca de um livro especifico até o controle dos empréstimos, o processo pode se tornar complicado a erros. Um sistema de gerenciamento digital facilita a busca por livros, a reserva e o acompanhamento de prazos de devolução, além de fornecer informações em tempo real sobre a disponibilidade do acervo.

Como por exemplo, um aluno que poderá acessar a plataforma digital para verificar a disponibilidade de um livro sem precisar ir até a biblioteca. Ele poderá reservar livros, e saber exatamente até quando pode mantê-los emprestados. Através do sistema automatizado, é possível registrar empréstimos e devoluções de maneira eficiente, além de gerar relatórios completos que auxiliam no controle do acervo e na tomada de decisões, como a necessidade de reposição ou aquisição de novos títulos.

Por sua vez é sobre os benefícios que esse sistema traz para toda a comunidade escolar. A acessibilidade é uma das principais vantagens: os alunos podem realizar suas consultas a qualquer momento, de qualquer lugar. O sistema também simplifica o trabalho dos bibliotecários, permitindo que dediquem mais tempo as outras atividades, como curadorias de novas obras e promoção de leitura.

**2 REQUISITOS DO SISTEMA**

Requisitos funcionais são descrições das funcionalidades e comportamento esperados de um sistema ou software. Eles definem o que o sistema deve fazer e as interações com os usuários, outras partes do sistema, ou ambos. Por exemplo cadastro de usuário, login e autenticação, buscar informações, atualizar dados, gerar relatórios, notificações e etc. no caso isso seria requisito funcionais.

**2.1 Requisitos funcionais**

Requisitos funcionais são especificações detalhadas que definem o comportamento de um sistema ou da própria aplicação. Eles explicam as funcionalidades que o sistema deve possuir para atender ás necessidades e expectativas dos usuários. Em outras palavras, os requisitos funcionais são as ações definidas que o software deve ser capaz de realizar.

Esses requisitos são fundamentais no desenvolvimento de software, pois servem como base para a criação, implementação e validação do sistema. Eles garantem que o produto final antedá aos objetivos para os quais foi projetado. Os requisitos funcionais podem incluir uma variedade de funcionalidades como, Autenticação do usuário, aonde o sistema permite que os usuários façam login, utilizando um nome de usuário e senha, Processamento de Transações, que o sistema é capaz de processar pagamentos e registrar transações financeiras, a Geração de Relatórios, que o sistema gera relatórios especificados sobre as atividades dos usuários ou sobre desempenho do sistema e a Geração de dados onde o sistema permite a atualização, exclusão, inserção e consulta de dados armazenados.

Para que os requisitos funcionais sejam eficazes, eles devem ser claros, específicos e mensuráveis. Isso significa que devem ser descritos de maneira que todos os envolvidos no projeto possam entender exatamente o que é esperado do sistema. Porém, devem ser verificáveis, ou seja, deve ser possível testar se o sistema atende a esses requisitos.

A criação de requisitos funcionais geralmente envolve a colaboração entre analistas de negócio, desenvolvedores, testadores, e os próprios usuários finais. Esse processo pode incluir entrevistas, workshops, análise de documentos e prototipagens. O objetivo é garantir reflitam fielmente as necessidades dos usuários.

Em suma, os requisitos funcionais basicamente são essenciais para o sucesso de qualquer projeto de software. Eles definem o que os sistemas devem fazer e servem como guia para todas as fases do desenvolvimento, desde o planejamento até a implementação e testes.

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

**2.2.1 RF004 – Cadastrar Aluno**

Descrição: O sistema deve permitir que o usuário cadastre novos livros no acervo da biblioteca.

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 deve verificar se os campos obrigatórios foram preenchidos

Saída:

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

**2.2.2 RF005 – Editar Aluno**

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

Prioridade: Alta

Entrada:

* Identificação do usuário que deve ser editado (ID ou RA).
* Campos a serem editados (ra, nome, sobrenome, data de nascimento, endereço, email, celular).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja algum tipo de informação que esteja errada ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações são salvas e o usuário é utilizado
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro.

Pós -condição:

* O aluno cadastrado deve ser cadastrado e seu usuário deve ser criado no sistema.

**2.2.3 RF003 – Excluir Aluno**

Descrição: O sistema deve permitir que o usuário seja excluído.

Prioridade: Média

Entrada:

* Identificação do livro a ser excluído (ID).

Processamento:

* O sistema deve verificar se há empréstimos ativos ou nome de usuário, caso exista a exclusão não é permitida.
* O sistema pede a confirmação de que o aluno quer excluir usuário.

Saída:

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

Pós-condição:

* O usuário deve ser removido do sistema.

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.

**DIAGRAMAS DO SISTEMA**

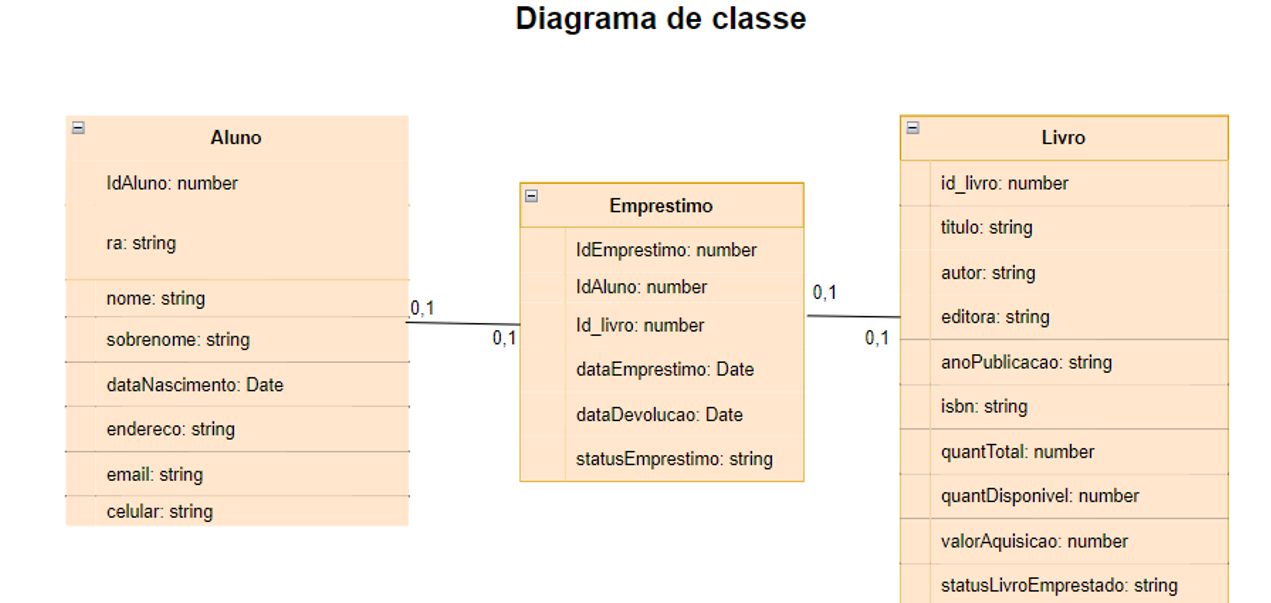
Um diagrama básico pode representar os principais componentes e suas interações. Um diagrama de casos de uso seria uma boa opção para visualizar as principais funcionalidades e como os usuários interagem com o sistema. No caso, diagrama de casos de uso da “Biblioteca” de exemplo seria, Aluno(Cadastrar Aluno, Editar Aluno e excluir Aluno).

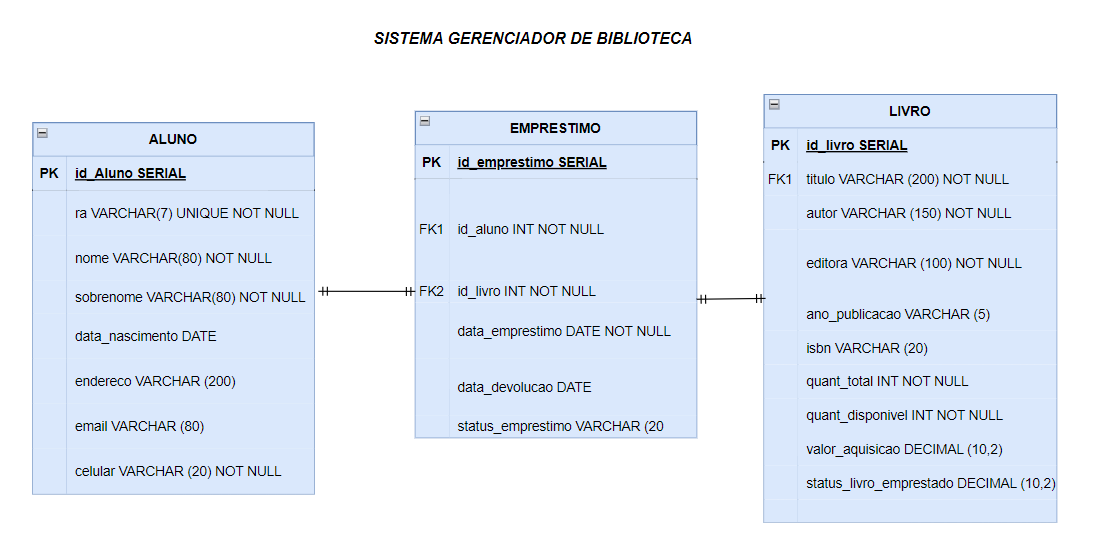
**3.1 Diagrama de Classe**

O diagrama de classe oferece uma visão clara da estrutura do sistema, mostrando com os objetos e suas respectivas classes estão inter-relacionadas. Ele ajuda da identificação de possíveis pontos de falhas ou áreas que precisam de ajustes, especialmente quando há mudanças no sistema, mas também ele permite organizar as funcionalidades de forma hierárquica e modular, facilitando a expansão do sistema.

O diagrama de classe server como um meio de comunicação eficiente entre a equipe de desenvolvimento, já que descreve visualmente a estrutura do código e as interações entre as classes.

Os diagramas de classe são uma ferramenta essencial no desenvolvimento de software. Eles representam a estrutura estática de um sistema, mostrando suas classes, atributos, métodos e os relacionamentos entre os objetos, e servem como uma linguagem como entre os membros da equipe, incluindo desenvolvedores, analistas de sistemas e stakeholders. Um diagrama de classe bem elaborada pode facilitar a comunicação e garantir que todos os envolvidos tenham uma compreensão clara do sistema.





**3.2 Banco de Dados**

O banco de dados é um sistema organizado para armazenar, gerenciar, e recuperar informações de maneira eficiente. Ele é essencial para diversas aplicações, desde sistemas empresarias até aplicativos móveis, suas características são, organizar e estrutura de dados, e esses dados são organizados em tabelas, que são compostas por linhas e colunas e cada tabela armazena informações sobre um determinado assunto, também sua capacidade de armazenamento, que podem armazenar grandes volumes de dados, permitindo a escalabilidade conforme a necessidade, segurança e privacidade, que implementam mecanismo de segurança para proteger os dados contra acessos não autorizados, confiabilidade e disponibilidade e disponibilidade, que garantem que os dados estejam disponíveis sempre que necessário, com mecanismo de backup e recuperação e a facilidade de acesso e manipulação, que permitem consultas rápidas e eficientes, além de operações de inserção, atualização e exclusão de dados.

O DER (Diagrama Entidade-Relacionamento) é uma ferramenta visual que representa a estrutura lógica de dados. Ele mostra as entidades (tabelas) e os relacionamentos entre elas. Importância do DER está em sua capacidade de facilitar o entendimento e o planejamento de banco de dados, garantindo que todas as relações e dependências sejam consideradas antes da implementação.

**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.

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

Rotas são um conceito fundamental que definem como as solicitações http são tratadas e direcionadas para diferentes partes de uma aplicação. As rotas são responsáveis por mapear URLs específicas para funções ou métodos que processam essas solicitações, permitindo que a aplicação responda de maneira adequada ás interações dos usuários, geralmente essas rotas são geralmente definidas em arquivos de configurações ou diretamente no código do servidor. Cada rota é associada a um caminho de URL e a um método http, por exemplo, uma rota pode ser configurada para responder a uma solicitação GET na URL /usuários e retornar a uma lista de usuários.

Um manipulador de rota é uma função que processa a solicitação recebida e gera uma resposta. Ele pode acessar dados do corpo da solicitação recebida e gera uma resposta. Ele pode acessar dados do corpo da solicitação, parâmetros de URL, cabeçalhos HTTP, etc. Por exemplo, um manipulador para a rota /usuários/:id pode extrair o id do usuário da URL e retornar os detalhes usuário específico.

Em aplicações maiores, as rotas são frequentemente organizadas em módulos ou controladores para manter o código limpo e gerenciável. Cada módulo pode lidar com um conjunto específico de funcionalidades, como rotas de usuários e de produtores.

As rotas são importantes para a estrutura de uma aplicação back-end. Elas permitem que a aplicação seja modular, escalável e fácil de manter. Além disse, uma boa organização de rotas melhora a segurança e a eficiência do sistema, garantindo que as solicitações dos usuários sejam tradas de maneira adequadas e segura.

**5 INTERFACE E EXPERIÊNCIA DO USUÁRIO**

A interface é o ponto de interação entre o usuário e um sistema, aplicativo ou dispositivo. Ela inclui todos os elementos visuais e interativos que permitem ao usuário navegar e utilizar o sistema, como botões, menus, ícones fontes e layouts. A principal função da interface de usuários é facilitar a interação, tornando-a intuitiva e eficiente.

Seu objetivo é nada mais e nada menos, que proporcionar uma experiência de uso agradável e eficiente, e isso envolve a facilidade de uso, quando a interface deve ser intuitiva, permitindo que os usuários realizem suas tarefas com o mínimo de esforço e sem a necessidade de um treinamento extenso.

O SGBD que utilizamos, foi o pgAdmin 4 v8, que oferece uma interface gráfica intuitiva para gerenciar o banco de dados, facilitando a execução da SQL, o monitoramento de desempenho e a criação de backups, permite que múltiplas operações sejam feitas simultaneamente, como abrir várias abas com diferentes consultas e essa interface torna a gestão do banco de dados PostgreSQL mais acessível tanto para administradores, quanto para desenvolvedores.

O Diagrama Entidade Relacionamento (DER) é essencial para visualizar as relações entre as entidades no banco de dados, garantindo um planejamento adequado do modelo de dados. Ele auxilia no design e estruturação de uma base de dados eficiente e sem redundâncias.

**5.2 Paleta de Cores**

Uma paleta de cores é uma seleção de cores escolhidas para serem usadas em um projeto de design. Ela é fundamental para criar uma identidade visual coesa e atraente, ajudando a transmitir a mensagem e os valores do projeto de forma eficaz. As cores escolhidas podem influenciar a percepção e as emoções dos usuários, tornando a experiência mais eficiente.

O objetivo dessas paletas é normalmente, harmonizar o design, garantindo que todas as cores usadas no projeto funcionem bem juntas, criando uma aparência visualmente agradável, guiar a atenção, pois, usar cores para destacar elementos importantes e guiar a atenção do usuário para as áreas mais relevantes do projeto/design e criar consistência, para mantes uma consistência visual em todas as partes do projeto, facilitando o reconhecimento e a familiaridade.

BIBLIOGRAFIA

Link das documentações utilizadas e livros consultados